

# 公立はこだて未来大学 2021 年度 システム情報科学実習 グループ報告書

Future University-Hakodate 2021 System Information Science Practice  
Group Report

## プロジェクト名

脳をつくるプロジェクト

## Project Name

Make Brain Project

## グループ名

視覚を持つ AI カー

## Group Name

AI car with vision

## プロジェクト番号/Project No.

20-B

## プロジェクトリーダー/Project Leader

石川慶孝 Yositaka Ishikawa

## グループリーダー/Group Leader

近藤裕貴 Yuki Kondo

## グループメンバ/Group Member

近藤裕貴 Yuki Kondo  
芥川裕大 Yudai Akutagawa  
青柳心吾 Shingo Aoyagi  
寺井遼大 Ryodai Terai  
坂中武蔵 Musashi Sakanaka  
折戸啓吾 Keigo Orito  
中嶋彪吾 Hyogo Nakasima

## 指導教員

香取勇一 佐々木博昭 ヴラジミール リアボフ 佐藤直行

## Advisor

Yuichi Katori Hiroaki Sasaki Volodymyr Riabov Naoyuki Sato

## 提出日

2022 年 1 月 19 日

## Date of Submission

January 19, 2022



## 概要

近年、人工知能に関する研究や開発が盛んに行われており、様々な場面で活用されている。その中で我々のグループは自動運転技術に注目した。自動車の普及に伴い、交通事故の件数も増加傾向にある。死亡事故も多発していることや高齢者ドライバーによる事故も増加傾向にあることから、大きな社会問題となりつつある。そして、その主な原因として運転手の確認不足や操作ミス、わき見運転といったヒューマンエラーが大半を占めている。自動運転技術によって運転手が自ら運転をする必要が無くなれば、そのようなヒューマンエラーによる交通事故の減少が期待できるため、近年注目を集めている。本グループでは、自動運転に用いられている機械学習や画像認識といった技術に触れ、「視覚を持つ AI カー」を成果物として作成することにより、自動運転について理解を深めることを目標とした。具体的には、ラジコンカーに Raspberry Pi4 を搭載し、PWM 制御によってモーターを制御する。そして、Python 言語を用いて実装したプログラムによって、カメラがとらえている状況を認識し、それに応じた走行ができる成果物の開発を目指す。

前期の活動では、扱う技術に関する基礎的な知識の習得や成果物を制作していくための環境構築、プログラミングの練習を行った。また、成果物に関する情報をグループ内で共有し、制作を進めていくために必要な Raspberry Pi4 やその周辺機器であるマウスやキーボードといった物品の選定も行った。後期の活動では、ラジコンカーの制御を行うグループと機械学習を実装するグループに分かれ、成果物の制作を進めた。ラジコンカーを担当するグループでは、ラジコンカーの組み立てや走らせるためのコース制作、PWM 制御の調整を行った。機械学習を担当するグループでは、成果物に用いる技術を選定した上で、画像認識を実装するためにカスケード分類器の作成や CNN モデルの構築を行った。前期の時点では、コース上の白線に沿って走行すること、コース上の障害物を避けられること、道路標識を認識し、その内容に応じた動作ができることを成果物の目標として設定していた。しかし、知識の習得や環境構築に想定していたより時間がかかってしまい、前期の間に成果物の制作に着手できなかった。そのため、成果物がどこまでの動作ができるようになるか後期の活動を進めていく中で、当初の目標を変更していった。前期に掲げた目標の内、コース上の白線に沿って走行するという目標についてはラジコンカーが自走するために最低限必要な機能であると判断したため、後期の活動が始まってから優先的に進めた。コース上の障害物を避けられるという目標については、ラジコンカーの制御を進めていくにあたって、難しい動作である可能性が感じられた。ラジコンカーが走行中に障害物を避けるには、障害物を認識した後、ぶつからないように一度停車し、避けられる方向を計算した上でその方向に曲がれるようにモーターを制御する必要がある。一つ一つの動作にどれくらいの期間を要するか目安を立てることができなかったため、コース上に道路標識以外の障害物を配置しないこととした。道路標識に応じた動作ができるという目標については、一つの道路標識のためのカスケード分類器を作成するのに多くの時間を要してしまった。十分な精度を持ったものを完成させるために、カスケード分類器に学習させる正解画像、不正解画像の枚数の調整を繰り返し行ったが、明確な改善策を中々見出せなかった。そのため、最初に作成を始めた「一時停止」の道路標識を認識するためのカスケード分類器しか完成させることができず、複数の標識を認識させるまで到達することはできなかった。したがって最終的には、白線に沿った走行、センサによる障害物の検出、カスケード分類器による「一時停止」の標識が認識できる成果物を作成した。作ったコース上でのみ走行が成功できたため、我々のグループの成果物ではあらゆる状況に対応できるわけではない。人の関与無しに全てを任せられる自動運転技術の開発には、当初想定していた機能をはじめ、さらなる改良が見込まれる。

**キーワード** 自動運転、画像認識、機械学習

(文責: 折戸啓吾)

# Abstract

In recent years, there has been a lot of research and development on artificial intelligence, and it is being used in a variety of situations. Among them, our group has focused on automated driving technology. With the spread of automobiles, the number of traffic accidents is on the rise. The number of fatal accidents is also on the rise, and the number of accidents involving elderly drivers is also on the rise, making it a major social problem. The main cause of these accidents is human error, such as inadequate checking by the driver, operational errors, and distracted driving. If automated driving technology eliminates the need for drivers to drive themselves, it is expected to reduce the number of traffic accidents caused by such human errors, and this has attracted much attention in recent years. In this group, we aimed to deepen our understanding of automated driving by creating a "visionary AI car" as a deliverable through exposure to machine learning and image recognition technologies used in automated driving. Specifically, we installed a Raspberry Pi4 in a radio-controlled car and controlled the motors using PWM control. The goal is to develop a program using the Python language that can recognize the situation captured by the camera and drive the car accordingly. In the first semester, the students acquired basic knowledge of the technologies to be used, built the environment for creating the deliverables, and practiced programming. We also shared information about the deliverables within the group and selected the Raspberry Pi4 and its peripherals, such as mice and keyboards, necessary to proceed with the production. In the second semester, we were divided into two groups, one for controlling the radio-controlled car and the other for implementing machine learning, and proceeded to create the deliverables. The group in charge of the radio-controlled car assembled the car, created a course for it to run on, and adjusted the PWM control. The group in charge of machine learning selected the technology to be used for the deliverables, and then created a cascade classifier and built a CNN model to implement image recognition. In the previous semester, we had set the following goals for the deliverables: to drive along the white lines on the course, to avoid obstacles on the course, and to recognize road signs and perform actions according to their contents. However, it took longer than expected to acquire the knowledge and build the environment, and we were not able to start creating the deliverables during the first semester. Therefore, as we proceeded with the activities in the second half of the year, we changed our initial goals to see how far the deliverables would be able to operate. Of the goals set in the previous year, we decided that the goal of driving along the white line on the course was the minimum necessary function for a radio-controlled car to drive itself, so we prioritized it after the activities in the second half began. As for the goal of being able to avoid obstacles on the course, we felt that it might be a difficult operation when we proceeded with the control of the radio-controlled car. In order for a radio-controlled car to avoid an obstacle while running, it needs to recognize the obstacle, stop once to avoid hitting it, calculate the direction to avoid, and then control the motor to turn in that direction. Since we could not estimate how long it would take to complete each operation, we decided not to place any obstacles other than road signs on the course. As for the goal of being able to perform actions according to road signs, it took a lot of time to create a cascade classifier for a single road sign. In order to complete a classifier with sufficient accuracy, we repeatedly adjusted the number of correct and incorrect images to be learned by the cascade classifier, but we could not find a clear way to improve it. As a result, we could only complete the cascade classifier for recognizing the "stop" road sign, which we started to create first, and could not reach the point where it could recognize multiple signs. Therefore, in the end, we created an artifact that could drive along the white line, detect obstacles using sensors, and recognize the "pause" sign using the cascade classifier. Since we were only able to successfully drive on the course we created, our group's artifacts cannot handle all situations. The development of automated driving technology, which can take care of everything without human involvement, will require further improvements beyond the originally envisioned functions.

**Keyword** Automated driving, Image Recognition, Machine Learning

(文責: 折戸啓吾)

# 目次

<b>第 1 章</b>	<b>はじめに</b>	<b>1</b>
1.1	背景 . . . . .	1
1.2	目的 . . . . .	1
<b>第 2 章</b>	<b>序論</b>	<b>2</b>
2.1	関連研究 . . . . .	2
2.2	具体的な課題設定 . . . . .	2
2.3	短期目標と長期目標 . . . . .	3
<b>第 3 章</b>	<b>方法</b>	<b>4</b>
3.1	提案手法 . . . . .	4
3.2	評価手法 . . . . .	15
<b>第 4 章</b>	<b>結果</b>	<b>16</b>
4.1	前期の結果 . . . . .	16
4.2	後期の結果 . . . . .	17
<b>第 5 章</b>	<b>考察</b>	<b>20</b>
5.1	前期 . . . . .	20
5.2	後期 . . . . .	21
<b>第 6 章</b>	<b>外部評価</b>	<b>22</b>
6.1	中間発表 . . . . .	22
6.2	成果発表 . . . . .	24
<b>第 7 章</b>	<b>まとめ</b>	<b>27</b>
7.1	前期、後期の取り組みについて . . . . .	27
7.2	成果 . . . . .	29
7.3	改善点 . . . . .	31
<b>第 8 章</b>	<b>個人の取り組み</b>	<b>33</b>
8.1	近藤裕貴 . . . . .	33
8.2	坂中武蔵 . . . . .	34
8.3	寺井遼大 . . . . .	36
8.4	中嶋彪吾 . . . . .	38
8.5	芥川裕大 . . . . .	39
8.6	青柳心吾 . . . . .	40
8.7	折戸啓吾 . . . . .	42
	<b>参考文献</b>	<b>44</b>

# 第 1 章 はじめに

## 1.1 背景

現在、全世界で年間 135 万人が交通事故によって亡くなっている。これは 24 秒に 1 人が亡くなっているという計算になる。そして、これらの殆どが人為的なミスが原因である。また、内閣府の出している警視庁資料による令和 2 年の「道路交通の動向」[1]によると日本で起こる事故も例外ではなく、交通事故のほとんどが追突や右・左折時衝突、出会い頭の衝突といった人為的ミスによるものである。こういった状況の中、自動運転技術がこれらの事故を減らす可能性があるという注目を集めている。また、高齢化が進んだ日本では高齢者による事故も多発しており、人口に対する交通事故死者数は、高齢者が全年齢の 2 倍となっている。そのため、日本では安全運転サポート車という衝突の危険時に自動でブレーキを作動させる・センサーにより前方の障害物の発見するような身体機能の低下を補う運転支援機能を搭載した車を用いて、交通事故の発生を抑制に努めている。自動運転には自動運転技術によって事故を減らせる上に、運転が難しくなった高齢者がサポート車よりも安全に自分の車で移動できるようになり、今後認められればさらなる高機能な自動運転車により、今より大幅な事故リスクの低減を狙えるという利点がある。また、自動運転車にはテスト車であるが周囲の状況をほぼ完璧に認識、動作を行える自動運転車であったり、信号情報を受け取りスムーズな運転を心がける車であったりするように現在、研究や調査の行われている車もある。このように、自動運転技術の実現、普及による恩恵は高齢者の方にはもちろんのこと、車を使うほとんどの人に対して多大であり今後も機械学習や画像認識において注目されていく技術であると考えられる。

(文責: 坂中武蔵)

## 1.2 目的

我々 AI カーグループは、背景で前述したように増加しているヒューマンエラーによって起こる交通事故を減らすために、機械学習や画像認識を用いて安全に自走できる AI カーを開発し、自動運転について理解を深めることを目的としている。そのために、私たちのグループでは AI カーに搭載する技術として、白線の認識・障害物の認識・標識の認識といった大きく分けて三つのものを検出・認識し、それぞれに対応した動作を行う機能の開発、搭載を目標としている。また、作成途中で新たに搭載が必要と考えられる機能が発見、考案された場合はそれらの機能を随時増やすことを考えている。

(文責: 坂中武蔵)

## 第 2 章 序論

### 2.1 関連研究

我々 AI カーグループが制作した制作物と似ているもので「Donkey Car」というものがある。これは、人間が操作して教師データを作成したものをを用いて学習させる人工知能を搭載した AI ロボットカーである。Google が提供している機械学習のためのソフトウェアライブラリである「TensorFlow」(テンサーフロー) を小型の CPU ボードの Raspberry Pi4 に搭載し、ラジコンカーにカメラを搭載して作るものである。この Donkey Car を走行させるまでの機械学習の流れは次の通りになっている。まず、作成したロボットカーをコントローラーで操作し、コース内を 5~10 周走らせる。この時、ロボットカーのカメラから得られる映像データや車輪を駆動するモーターの回転数、コントローラーのハンドルから得られた前輪の舵角データを集める。その後、PC 上で Deep Learning(Tensorflow) を用いて記録した大量のデータを学習させる。そして、学習済みのモデルやプログラムを Raspberry Pi4 に入れ、ラジコンカーのカメラの映像(外からの情報)に対してモーターの回転数、ハンドルの角度をコントロールして自動走行を可能にするという仕組みである。この既存の物では、人間がロボットカーを操作し同じコースを何周も走らせる必要がある。また、同じコースを何周も走らせ学習させるため、複雑なコースでは対応できずコースアウトしてしまう可能性が高い。

そのため、我々 AI カーグループは、人間の手を一切加えずにどんなコースでも対応して走行することができる AI カーの作成を試みた。コースを学習させずに簡単なコースから複雑なコースまでをコースアウトすることなく自走するためには白線認識が必要になる。また、今回の目的である安全に自走が可能という点から障害物の検知や標識の認識を可能とした AI カーの作成を行った。

(文責: 中嶋彪吾)

### 2.2 具体的な課題設定

我々のグループでは、Raspberry Pi4 にカメラモジュールを搭載し画像認識を用いてリアルタイムで周囲の状態を認識し機械学習により障害物か標識かを識別することで動きを制御することを行う。ここで必要なのが障害物と標識を識別するためのデータセット、機械学習、画像認識である。プログラムはカメラモジュールからリアルタイムに周囲の画像を認識し、障害物か標識かを識別させ、PWM 制御により Raspberry Pi4 からモーターへ信号を送ることで AI カーを制御する。また、障害物・標識だけでなく同時に白線も認識し、動作を行うため障害物・標識・白線を同時に認識の行えるカメラモジュールは必須となり、白線認識と障害物・標識の認識を同時に処理、制御の行えるプログラムも必要となる。以上の観点から、障害物・標識・白線の認識に用いることのできるライブラリの種類やそれらの認識を同時に処理、制御を行える等の理由により、開発言語は Python を用いることとした。データセットには障害物と標識の画像データが必要となり、それぞれの画像と機械学習を用いた画像認識のプログラムが必要である。

したがって、我々のグループの課題は画像のデータセットの収集と識別のための機械学習を用いた画像認識のプログラムと障害物・標識・白線の認識を同時に行う事のできるカメラモジュールの

調査、入手と障害物・標識・白線の認識を同時に行う事の出来るプログラムと画像認識の結果から PWM 制御を利用したモーターの制御である。

(文責: 坂中武蔵)

## 2.3 短期目標と長期目標

短期目標として、前期には前節にもあるように使おうと考えている技術の知識を深める、開発環境の決定、構築を並行して行っていく。また、「画像認識プログラミングレシピ」[2] や用いるプログラムの問題を解くなどをして学びを深める。最終成果物にどのような機能を搭載するかについても考えていく。具体的には、「画像認識プログラミングレシピ」による学修や担当教員である香取教授の協力によるライブラリの学習で最適な方法の模索を行い、今回のプロジェクトで用いる技術について学び、私たちのグループが実現を考えている機能の搭載において最良である技術を考える。また、学習した内容や搭載方法などから開発環境を決定しその環境での最終成果物に搭載する機能の実現方法を考える。

前期時点での長期目標としては、AI カーのサンプルとして用いる Picar-X の作成、解析及び最終成果物に用いる様々な物品の調査、購入である。本グループの最終成果物には白線・障害物・標識の三つのものを検出・認識する機能が必須となる。AI カーに搭載する機能に必要な部品や搭載する機能の動作等を確認、調査するために最終成果物のモチーフとなるような機能を持つサンプルを作成し、最終成果物の完成を進める。

後期の短期目標としては、最終成果物とする AI カーの外身の作成、画像認識・機械学習等を用いた各機能の作成・搭載などを行っていく。具体的には、前期の長期目標で定めた AI カーの解析、最終成果物に用いる物品の調査等により、目標としていた白線・標識・障害物の三つを認識し、自走する AI カーを完成させるために必要なことを行う。この際、作業が多岐にわたると予想されるため複数の組に分割して作業を行う事で効率良く目標の達成及び、最終成果物の機能の向上等を行えるように努める。

後期時点での長期目標については、最終成果物の完成である。具体的には、自分たちで作成したコースを自走して、定めた三つのものを認識できる機能を搭載した AI カーを最終成果発表日まで完成させることである。この時、コースは平坦で周回できる楕円形のを想定している。また、他に必要な機能が出てきたり、搭載を考えている機能の向上が見込めたりした場合、その都度成果物を更新していくと考えている。

(文責: 坂中武蔵)



## 第 3 章 方法

### 3.1 提案手法

#### カスケード分類器

物体検出を行うためには様々な手法が存在する。例えば、SSD や YOLO などが有名である。しかし、それらは動作させるコンピュータに GPU が必須である。私たちが利用した Raspberry Pi4 は GPU が組み込まれておらず SSD や YOLO と呼んだ物体検出アルゴリズムをしようすることが不可能であった。そこで、物体の領域の判定とその領域が何であるかを別々に学習させるという手法にした。物体の領域を判定させるために OpenCV の顔認証などにも利用されているカスケード分類器の利用をすることにした。

カスケード分類器は SSD や YOLO などの深層学習とは違い、あらかじめ決められた大量のパターンを様々な学習画像に対して当てはめ、画像の輝度変化などを特徴として抽出する方法である。

#### 特徴量

今回利用した OpenCV では三種類の特徴量識別器が存在する。一つ目は Haar-like 特徴量である。Haar-like 特徴量とは複数の局所的な明暗差の組み合わせにより判別する方法であり、Integral Image と呼ばれる手法 [3] を使用することで特徴量を高速計算することができる。二つ目は LBP (Local Binary Pattern) 特徴量である。LBP 特徴量とは画像の局所的な輝度の分布の組み合わせにより判別する方法 [4][5] であり、輝度変動に頑強で高速に計算できる。デメリットとしてスケーリングや回転などの幾何学的な変動には頑強ではないことが挙げられる。三つ目は HOG (Histograms of Oriented Gradients) 特徴量である。HOG 特徴量とはグレースケール画像の局所的な輝度の勾配方向をヒストグラム化し、そのヒストグラムを特徴量として判別する方法 [6] であり、勾配を特徴量としているため物体の形状を捉えるのに有効な特徴量とされている。対象の物体が一定の枠内に収まっている画像等では、照明の変化などにも強い。この三つの特徴量の中でも最も顔認証に利用されており精度が高い [7] と言われている Haar-like 特徴量を利用した。

#### 使用画像

カスケード分類器を作成するにあたりポジティブ画像とネガティブ画像が必要となる。ポジティブ画像 (生まれ標識) はインターネットから 10 枚収集し `opencv_createsamples` で水増しを行い増やした。ネガティブ画像は生まれ標識以外とし、機械学習コンペティションプラットフォームである Kaggle から収集した。ポジティブ画像とネガティブ画像の比率も重要になると考えたため、それぞれの画像の比率を変更させ様々な比率のカスケード分類器を作成した。

(文責: 寺井遼大)

## 画像の水増し・ベクトルファイル作成

カスケード分類器を作成するにあたりプログラム言語は Python を利用し、ライブラリである OpenCV の `opencv_createsamples` と `opencv_traincascade` を利用した。

### • `opencv_createsamples`

-info collection\_file\_name ... 正解画像のリストファイル（自力で集めた場合）  
-img image\_file\_name ...1 枚の画像（変形させてベクトルファイルを作成する場合）  
-vec vec\_file\_name ... 出力されるベクトルファイルの名前（～.vec）  
-bg background\_file\_name ...-img の背景になる画像のリストファイル  
-num number\_of\_samples = 1000 ... 作成する正解画像の枚数  
-bgcolor background\_color = 0 ... 正解画像の背景になっている色を指定  
-inv -randinv... 色の反転をする場合に指定  
-bgthresh background\_color\_threshold = 80 ... 背景とみなす範囲  
-maxidev max\_intensity\_deviation = 40 ... 最大明度差  
-maxxangle max\_x\_rotation\_angle = 1.100000 ... 最大回転角度 rad  
-maxyangle max\_y\_rotation\_angle = 1.100000  
-maxzangle max\_z\_rotation\_angle = 0.500000  
-show scale = 4.000000 ... 指定すると変形させた正解画像を確認できる  
-w sample\_width = 24 ... ベクトルの横幅  
-h sample\_height = 24 ... ベクトルの高さ

### • `opencv_traincascade`

-data cascade\_dir\_name ... 分類器を保存するフォルダ（分類器ごとに分ける）  
-vec vec\_file\_name ... 作成したベクトルファイル  
-bg background\_file\_name ... 不正解画像のリストファイル  
-numPos number\_of\_positive\_samples = 2000 ... 作成した正解画像の枚数  
-numNeg number\_of\_negative\_samples = 1000 ... 用意した不正解画像の枚数  
-numStages number\_of\_stages = 20  
-precalcValBufSize precalculated\_vals\_buffer\_size\_in\_Mb = 256 ... 特徴量用のメモリサイズ  
-precalcIdxBufSize precalculated\_idx\_buffer\_size\_in\_Mb = 256 ... 特徴量用のバッファメモリー

### • `cascadeParams`

-stageType BOOST(default)  
-featureType {HAAR(default), LBP, HOG } ... 特徴量の見つけ方  
-w sampleWidth = 24 ... 正解画像を作成したときと同じものを指定  
-h sampleHeight = 24 ... 正解画像を作成したときと同じものを指定

• **boostParams**

-bt {DAB, RAB, LB, GAB(default)} ...boost 分類器のタイプ

DAB - Discrete AdaBoost,

RAB - Real AdaBoost,

LB - LogitBoost,

GAB - Gentle AdaBoost.

-minHitRate min\_hit\_rate = 0.995 ... 各ステージでの最小ヒット率

-maxFalseAlarmRate max\_false\_alarm\_rate = 0.5 ... 各段階で最大の誤認識率

opencv\_createsamples と opencv\_traincascade は OpenCV バージョン 3 以下でしかインストールできないため、OpenCV バージョン 3.4.16 をインストールした。opencv\_createsamples とはポジティブ画像の角度、明度を変更させ水増しさせると同時にベクトルファイルに変換させるアルゴリズムである。カスケード分類器を作成する方法として 2 パターンの方法を試した。ポジティブ画像の角度、明度を変更させ水増しさせベクトルファイルを作成する方法を手法 A、ポジティブ画像の角度、明度を変更させ背景を付け足し水増しさせベクトルファイルを作成する方法を手法 B とした。手法 B は検出したい画像がどこにあるかの情報（アノテーション）をテキストファイルで作成することができるため精度の向上が見込める。手法 A、手法 B で使用するネガティブ画像はテキストファイルを作成し、その中にネガティブ画像の名前をリストとして記述し使用できるようにした。手法 A では、収集していた 10 枚のポジティブ画像をそれぞれ 1,000 枚に水増しさせ、合計 10,000 枚のポジティブ画像を作成しベクトルファイルで保存した。opencv\_createsamples をしようする際に、画像の回転角度を変更する maxxangle、maxyangle、maxzangle は 0.8、0.8、0.5 とした。作成した 10,000 枚のポジティブ画像の中からランダムにカスケード分類器の作成に必要な枚数を選ぶこととした。手法 B では、ポジティブ画像とネガティブ画像とは別に背景画像を Kaggle から 10,000 枚収集し利用した。背景画像はネガティブ画像と同様にテキストファイルを作成し、その中に背景画像の名前をリストとして記述した。収集した背景画像はポジティブ画像とは無関係の画像であり、ネガティブ画像とは重複しないようなものを選んだ。手法 A と同様に収集していた 10 枚のポジティブ画像をそれぞれ 1,000 枚に水増しさせ、合計 10,000 枚のポジティブ画像を作成した。画像の回転角度を変更する maxxangle、maxyangle、maxzangle は 0.8、0.8、0.5、水増しした 10,000 枚のポジティブ画像は全て背景の異なる画像である。手法 A 同様、作成した 10,000 枚のポジティブ画像の中からランダムにカスケード分類器の作成に必要な枚数を選ぶこととした。手法 A と手法 B で作成したベクトルファイルからカスケード分類器を作成した。カスケード分類器の作成には opencv\_traincascade の vec に作成したベクトルファイル名を記述しベクトルファイルを指定する。学習するステージ数も変更でき今回は 20 ステージとした。精度が高いカスケード分類器を作成するため、ポジティブ画像とネガティブ画像の比率を重視した。まず、ポジティブ画像を 2,000 枚に固定しネガティブ画像の枚数を変更することで比率を明確にした。比率を 1:0.5、1:1、1:2 としてネガティブ画像が 1,000 枚を分類器 A、2,000 枚を分類器 B、4,000 枚を分類器 C とした。テスト画像は止まれが連続している画像を使用し、何個認識できているかで精度を見極めた。

(文責: 寺井遼大)

## CNN モデル

CNN はカスケード分類器により検出された標識を判定するモデルとして使用した。CNN とは「Convolutional Neural Network」の頭文字を取ったもので、日本語では「畳み込みニューラルネットワーク」とも呼ばれる。脳のニューロンの仕組みを利用したニューラルネットワーク (NN) と呼ばれる機械学習手法の一種であり、NN では 3 層のノードからなり、層間のノード同士の重みを学習する全結合層のみで構成されているが、CNN ではこれらに加えて畳み込み層やプーリング層などの画像の幾何構造に対応する特徴を抽出するための層も用いてネットワークが構築されている。今回はクジラ飛行機、杉山陽一、遠藤俊輔「Python による AI・機械学習・深層学習アプリの作り方 TensorFlow2 対応」2020 年 10 月 21 日 [8] を参考に CNN モデルを作成した。

判定する標識は止まれのみであったため、正解画像を止まれ標識、不正解画像を止まれ標識以外とした。枚数は正解画像を 3960 枚、不正解画像を 3960 枚である。そして、データの 8 割をランダムに分割したものをトレーニングデータ、残りの 2 割をテストデータとして、500 回の学習を行わせた。以下にトレーニングデータとテストデータの精度を表すグラフを図 3.1 の a に示す。またトレーニングデータとテストデータの誤差を表すグラフを図 3.1 の b に示す。学習の結果を表すグラフを示す。

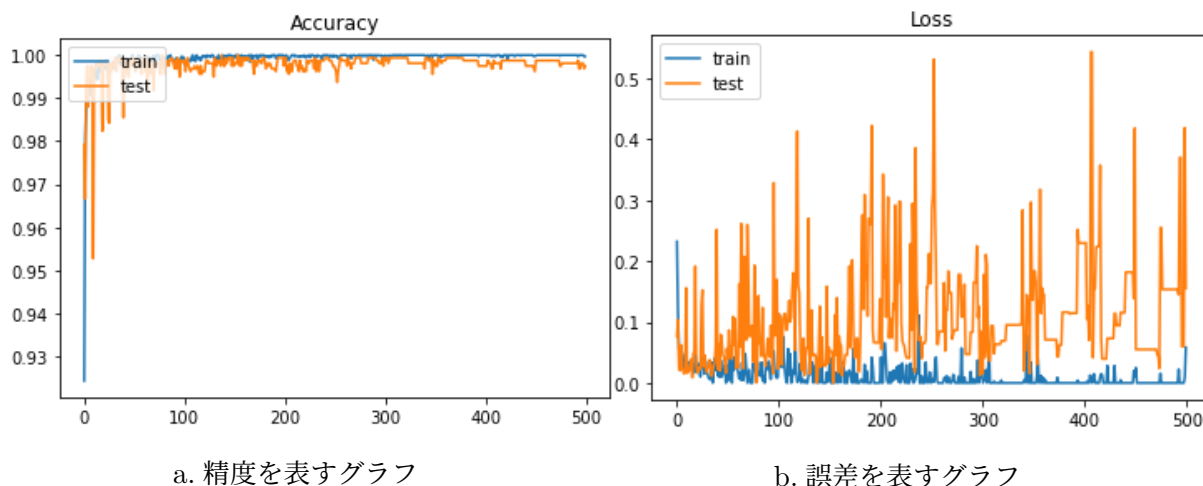


図 3.1

このモデルをもとに、Raspberry Pi4 に導入し検証しようとしたが、モデルに必要なモジュールを Raspberry Pi4 にインポートできなかった為、検証する事ができなかった。

(文責: 青柳心吾)

## 白線認識

白線認識では、カメラで取得した画像から2本の白線を検出し、その中心からのズレを求める白線検出を実装した。また、PID制御を用いることでそのズレからステアリングの角度を制御し、2本の白線の間を走らせることが出来た。

### <白線認識の方法と手順>

#### 1. カメラから画像を取得

一般的な Raspberry Pi4 用カメラモジュールでは白線全体を捉えることが出来ず、白線認識が困難であったため、広角レンズ付きの Raspberry Pi4 用カメラモジュールを使用して画像を取得した。

#### 2. Raspberry Pi4 の性能にあわせて画像の画質を下げる

カメラから取得した画像は、そのままの画質だと5秒ほどの遅延が発生する上に、1秒間に1枚ほどしか処理できなかった。画質を下げて処理速度が上がるとより正確な白線認識が可能になるが、画質を下げすぎると白線認識自体がうまく機能しなくなるため、白線認識が正しく機能する中でなるべく画質を落とした。

#### 3. 関心領域 (ROI:Region Of Interest) の抽出

取得した画像には余分な領域が多く、これが残っていると白線の誤検出の原因になるため、白線認識に必要な余分な領域を切り取った。今回は、画像の上から40%は常に地面より上の領域になっており、白線認識には必要ないため、それを切り取った。

#### 4. 逆透視投影 (IPM:Inverse Perspective Mapping) 変換

地上から撮影した画像を、上から見下ろした形に変換した。これによってその後の処理がしやすくなった。変換した画像が上から見下ろした形になっているかどうかは、2本の平行な直線を地上から撮った場合に変換後の直線が水平になっているかどうかで判定した。

#### 5. 白色抽出

画像から、白線に使われている色に近い色だけを抽出した。

抽出する白色のうち、完全な白色 (RGB:255,255,255) を除外することで、太陽や照明の反射光の誤検出を減らすことが出来た。

#### 6. 二値化

白線がきれいに浮き上がるように閾値を調整し、白と黒に2値化した。

#### 7. 白線を検出する

左右の白線を構成する白色のピクセル座標を求めた。具体的には、列ごとに白色ピクセルが連続する部分を2箇所取得し、それを縦に繋げたものを白線とした。[図 3.3]

(文責: 芥川裕大)

## <PID 制御の方法と手順>

### 1. 中心線を求める

左右の白線から等感覚な位置の座標を求め、それらを縦に繋げたものを白線の中心線とした。

### 2. レーンの中心とのずれを求める

今回は検出した白線の中心線の中から一点を選び、その点と画像の下辺の中心点を結んだ線と、画像の中心線との角度を求め、その角度をずれの値とした。白線の中心線の中から選ぶ1点は、車体に近いほどステアリング制御の反応が遅れるため、カーブが曲がりきれなくなり、車体から遠いほどステアリング制御の反応が早いため、カーブの内側に入った。また、車体のスピードにも左右されるため、スピードに合わせてその都度調整を行った。

### 3. ずれの値からステアリングの角度を決める

P 制御と PI 制御の2種類の制御方法を使ってステアリングの角度を求めた。PID 制御についても検討したが、時間的問題から実装することはできなかった。

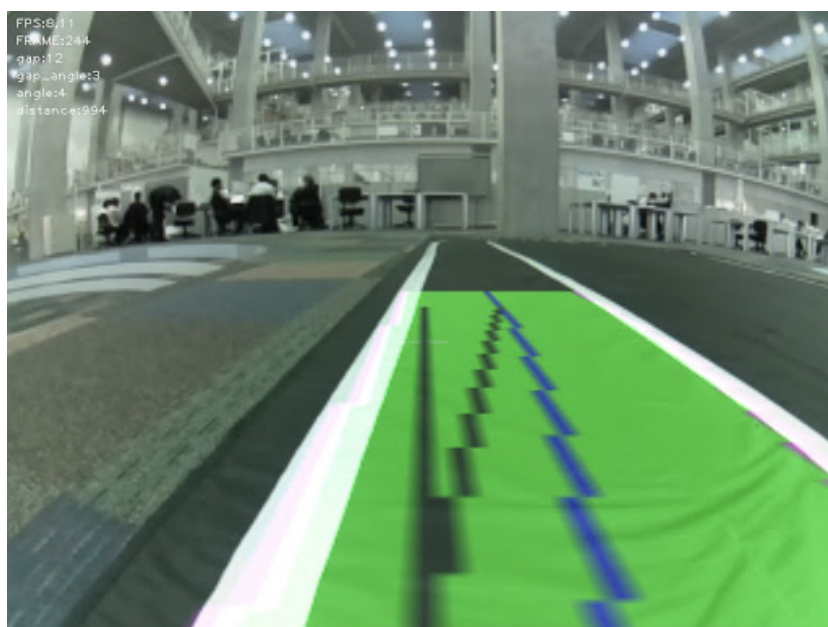


図 3.2 白線を検出し、レーンの中心からのズレを求める様子

## <PID 制御とは>

Proportional-Integral-Differential Controller の略称で、温度制御などの各種制御に用いられる制御方法。入力値の制御を、現在値と目標値との偏差と、その積分、微分を用いて行う。具体的には、現在値と目標値との偏差に比例した出力を出す比例動作、その偏差の積分に比例した値を出す積分動作、その偏差の微分に比例した値を出す微分動作の三つの要素の和を出力値とする。

状況によって、比例動作だけを用いた P 制御、比例動作と積分動作を用いた PI 制御を用いる場合もある。[9][10]

(文責: 芥川裕大)

### <白線認識における P 制御と PI 制御>

今回の白線認識では、比例動作と積分動作を求め、P 制御、PI 制御を使って AI カーのステアリングを制御した。比例動作については、AI カーとレーン中央とのズレの値に、” 比例ゲイン” をかけた値とした。積分動作については、AI カーとレーン中央とのズレの値を” 積分時間” だけ累積し、その累積値に” 積分ゲイン” をかけた値とした。

P 制御では、比例動作の値をそのままステアリングの角度に反映した。また、AI カーを実際に走らせながら” 比例ゲイン” の値を少しずつ変えていき、レーンの中心を滑らかに走れるような” 比例ゲイン” の値を調べた。

PI 制御では、比例動作と積分動作の和をステアリングの角度に反映した。また、AI カーを実際に走らせながら” 比例ゲイン”、” 積分ゲイン”、” 積分時間” の値を少しずつ変えていき、レーンの中心を滑らかに走れるようなこれらの値を調べた。

### <白線認識における PID 制御の必要性>

#### ・ P 制御

P 制御を使わずに、車体がレーン中央より右側にあるときはステアリングを左向きにし、車体が左側にあるときはステアリングを右向きにするという簡単な方法でも、レーンの中央に沿って走ることが可能である。しかし、このやり方では車体がレーン中央に戻っても、車体が中央に向かって斜め横を向いた状態のままであるため、レーン中央を超えて行き過ぎてしまう。その結果、中心線の周りを行ったり来たりしながら前に進んでしまう。

このような場合に P 制御を用いることで、滑らかにレーン中央を走ることができる。具体的には、車体とレーン中央との距離に比例した値をステアリングの値にする。その結果、車体がレーン中央から離れているほどステアリングが中央側を向き、車体がレーン中央に近づくほどステアリングが正面を向くので、車体がレーン中央を超えて行き過ぎてしまうことなくレーンの中心を走ることができる。

#### ・ PI 制御

P 制御を用いた場合、現在値が目標値に近づくほどステアリングの値が小さくなってしまふ。そのため、オフセット（残留偏差）と呼ばれる目標値と現在値とのズレが一定の値で永続的に続いてしまふ現象が発生し、目標値に完全に返ることが出来なくなってしまう。

このような場合に積分動作を加えることで、それを修正することができる。具体的には、車体とレーン中央との距離を加算していったものに比例した値（積分動作）を、比例動作の値に足すことで、オフセットが続いた場合に積分動作が大きくなり、オフセットをなくすことができる。

#### ・ PID 制御

PI 制御を用いた場合、外的要因で現在値が大きく目標値から離れた場合に、I 動作はある程度時間が経たないと働かないため、その修正に時間がかかる。

このような場合に微分動作を加えることで、迅速にそのズレを修正することができる。

（文責：芥川裕大）

## <AI カーの制御・構造>

### ・AI カーの制御

AI カーのスピード制御や前輪のステアリング・操舵角の制御は PWM 制御を用いてそれぞれ制御を行っている。まず、PWM 制御とはどのようなものなのか解説し、スピード制御やステアリング制御についての詳細を説明する。

### ・PWM 制御とは

PWM (Pulse Width Modulation) とは、半導体を使い高速スイッチングにより電力を制御する方法である。繰り返しスイッチのオンとオフを行い、出力される電力を制御することを PWM 制御という一定電圧の入力から、パルス列のオンとオフの一定周期を作り、オンの時間幅を変化させる電力制御方式を PWM と呼ぶ。この制御方式を PWM 制御と呼ぶ。PWM 制御は、優れた制御性と高効率が特徴でインバータ回路など広く使われている技術の一つである。そのため、ブラシ付き DC モーター制御、オーディオアンプなどの分野にも広く使われ、モーター制御の場合は、インバータ回路で PWM 制御のオンの時間幅 (デューティ比) を連続で変化させることでモーター駆動に最適な理想に近い正弦波の交流電流を作ることができる。[11]

### ・Raspberry Pi4 を用いた PWM 制御

今回、我々は、Raspberry Pi4 で PWM 制御を行うため、PCA9685 (サーボモータードライバ) を用いて制御を行った。この PCA9685 は Raspberry Pi4 から指示された波形の PWM 信号を発生させ、AI カーの操舵角を決定するステアリングモーターと駆動用モーターである DC モーターそれぞれに信号を送り、制御してくれる仕組みである。このサーボモータードライバを Raspberry Pi4 で扱うには PCA9685 専用のライブラリ「Adafruit python PCA9685」を予めインストールする必要があった。また、この 2 つの接続方法に I2C (Inter-Integrated Circuit) 通信を用いた。

I2C 通信とは、クロックに同期させデータの通信を行う同期式シリアル通信で、クロック (SCL)、データ入出力 (SDA) の 2 本の信号線を用いて通信する方法である。そのため、Raspberry Pi4 の GPIO ピンも I2C 通信専用の SDA と SCL をそれぞれサーボモータードライバに接続した。サーボモータードライバは 0 番ピンにステアリングモーター、1 番ピンにスピードコントローラーを接続することでそれぞれステアリング制御やスピード制御を行った。

### ・スピード制御

スピード制御は、スピードコントローラーによってブラシレスモーターが回転する PWM 値が異なる。そのため、PWM 値を少しずつ変えながらモーターを動かすことができる範囲を見つける必要があった。今回用いたスピードコントローラーでは、PWM 値を 280 から 356 の間にすることで制御をすることが可能だった。PWM 値が 357 以上ではブラシレスモーターの回転は止まり、356 から値を小さくしていくことで回転速度を上げることができた。PWM 値が 356 は止まるか回転するか絶妙な速度で回転をし、280 でブラシレスモーターの回転速度は最大となった。今回、AI カーの速度はコースが直進の時に PWM 値 355、曲がるときに 356 を用いてスピード制御を行った。



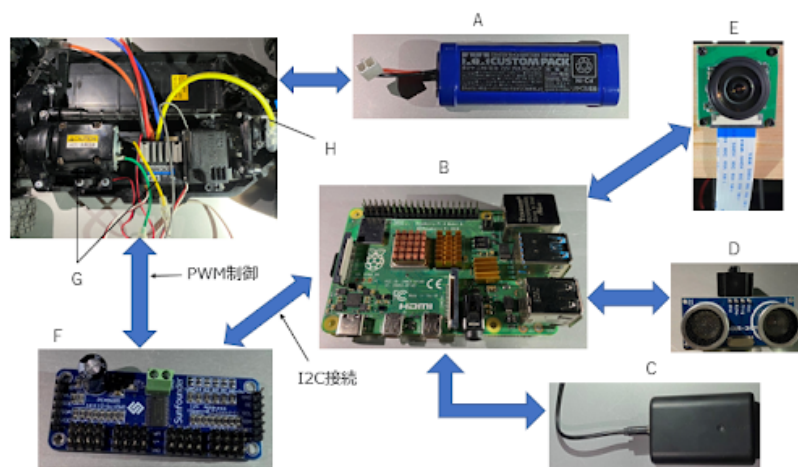
• ステアリング制御

ステアリング制御も、スピード制御と同様、PWM 制御を用いて制御を行った。今回、PWM 制御を行うために用いたライブラリでは操舵角に対する PWM 値が統一されていたため簡単に制御を行うことができた。しかし、前輪の可動域外の PWM 値を与えてしまうと前輪を制御するステアリングモーターが壊れてしまうため、可動域に対する PWM 値を調べる必要があった。前輪の可動域に対する PWM 値を調べたところ 350 から 450 の間で前輪が動くことが分かった。PWM 値 400 では前輪はほぼ真っ直ぐ向き、350 では前輪が可動域限界の左に切れ、450 では前輪が可動域限界の右に切れた。この AI カーのステアリング制御と白線認識、PID 制御を同時に用いることで自走させることが可能となった。

(文責: 中嶋彪吾)

## <AI 車の構造について>

我々が作成した AI 車の中身や構造は、関連研究でも説明した「Donkey Car」を参考にした。これから、AI 車の構造について説明していく。



A. AI 車のスピードコントローラーとブラシレスモーターを動かすためのバッテリー。Raspberry Pi4 からの電源供給では電力不足でスピードコントローラーを動かすことができないので、タミヤの充電式ニカド電池を使用した。

B. Raspberry Pi4。Raspberry Pi4 は AI 車の心臓部分にあたり、ステアリングモーターやスピードコントローラー、カメラ、超音波センサなど様々な機器に指示を送る働きをする。

C. モバイルバッテリー。Raspberry Pi4 に電源供給を行うために用いた。

D. 超音波センサである。超音波センサは AI 車の前に備え付けられており、障害物との衝突を回避するために用いた。

E. カメラモジュール。このカメラは Raspberry Pi4 と接続されていて、白線認識や標識認識を行うために用いた。

F. サーボモータードライバ (PCA9685)。この、モータードライバはスピードコントローラーとステアリングモーターに PWM 信号を送ることができ、スピード制御とステアリング制御を同時に制御できるため使用した。また、Raspberry Pi4 で PWM 制御と GPIO 制御を行うには、様々なライブラリ (pigpio, WiringPi, RPi.GPIO など) を同時に処理するプログラムが必要になる。しかし、このサーボモータードライバを動かすためのライブラリ (Adafruit Python PCA9685) を用いることで 2 つの制御を同時に簡単に制御することができるようになる。

G. スピードコントローラー (ESC) とブラシレスモーター。このモーターは、スピードコントローラーなしで動かすことができないため、PWM 制御でスピードコントローラーを制御しモーターの回転数を調整している。

H. ステアリングモーター。このモーターは、PWM 制御を用いることで動かすことができ、PWM の値によって前輪の向きを調整し制御している。

(文責: 中嶋彪吾)

## <物体検出>

我々のグループでは障害物の物体検出に「超音波センサ」を用いた。この章では、いろいろある技術やシステムの中でなぜ超音波センサになったのかという経緯と超音波センサについての詳しい仕組みや詳細を説明する。

### 経緯

我々は、現在の自動運転技術に用いられる障害物を検知する技術についてどのようなシステムがあるか無知であった。そのため、インターネットを使い調べることで、物体検出に使われる技術やシステムの知識を増やした。調べたところ、自動運転の物体検出を行う方法として、以下の技術が使われていることが分かった。

ステレオカメラ：人が物を見る原理と同じように、2つのカメラ(2眼のカメラ)を用いて対象物を複数の異なる方向から同時に撮影することにより、カメラの画素の位置情報から、奥行き方向の情報も計測することが可能なカメラ。[12]

3D-LiDAR：3次元状に光を照射し、センサ周辺の距離からその場所の形状やセンサ位置の把握に活用されるセンサである。光を照射して距離を計測する LiDAR センサが機械的に回転して、360度の全周囲の距離計測が可能になる。[13]

ミリ波レーダー：電波を飛ばして障害物を検知する技術を活用したセンサである。ミリ波帯(波長が1~10mm = 周波数 30~300GHz)の電波を使うため、ミリ波レーダーと呼ばれている。[14]

超音波センサ：超音波を利用した非接触型のセンサで、物体の有無や物体までの距離を検出することが可能である。[15]

現在の自動運転で使われている4つの技術から実際に Raspberry Pi4 を用いて作る AI カーに実装可能であるものを考えた。まず、一つ目のステレオカメラでは、白線認識用のカメラを1つ用いる予定だったため、2つ目のカメラを用いることで FPS 値が低下し標識認識や白線認識を正確に行わない可能性があったため却下となった。2つ目の 3D-LiDAR は、自動運転に使われているだけあり高価なもので買うことができないという理由と Raspberry Pi4 に対応したものがなさそうという理由から用いないことにした。3つ目のミリ波レーダーは4つ目の超音波センサと仕組みが似ていることから、Raspberry Pi4 でも FPS 値を下げることなく AI カーに実装することができる可能性があるということで超音波センサを用いることにした。

(文責: 中嶋彪吾)

## 仕組み

超音波センサは、「HC-SR04」を使用した。超音波センサは、超音波を利用した非接触型のセンサで、物体の有無や物体までの距離を検出することができる。一般的に超音波振動子（圧電素子）を備えた発振部と受信部から構成されており、発振部から発振された音波を直接、もしくは反射させて受信部で受信し、対象物を検出する。超音波センサには透過型、回帰反射型、物体反射型の3種類あり、それぞれハウジングの形状や設置方法、検出方法が異なる。今回は回帰反射型超音波センサを用いて物体検出を行った。物体反射型は、超音波センサの中で最も多く使用されているタイプで、非接触で対象物までの距離や対象物の有無を検出することができる。発振部と受信部の両方の機能を備えた発振受信部を、検出したい方向へ向けて配置する。今回、我々の作成した AI カーはコース上での障害物があるかないかを想定したため AI カーの真正面に超音波センサを取り付けた。コース上に障害物が存在しない時は、発振受信部から発振された音波は、そのまま散乱し、受信部に音波は帰ってこない。しかし、コース上に障害物がある時、対象物が発振受信部の前に進入すると、対象物に音波がぶつかり反射し、反射した音波を発振受信部で受信する仕組みになっている。この音波が発振してから戻ってくるまでの時間により、対象物までの距離や有無を検出し、距離はアナログ（電流値、若しくは電圧値）、有無はデジタル（ON/OFF）で出力する。[16]

## 特徴

超音波センサの特徴は大きく3つある。まず、1つ目に、「透明体の検出が可能」という点である。そのため、ガラスや液面でも超音波を反射させることができるので様々な障害物を検知することが可能である。2つ目は、「ミストや汚れに強い」という点であり、障害物が多少汚れていても、影響を受けずに検出することが可能である。3つ目に「複雑な形状でも検出可能」である点である。そのため、細い柱やあみあみ上のフェンスなど様々な形状でも検出が可能である。

（文責: 中嶋彪吾）

## 3.2 評価手法

我々が作成した AI カーの完成度がどのくらいであるかを評価する。

初めに、白線認識の精度に注目する。具体的に3つの観点から評価する。1つ目は、白線認識自体ができていのかである。2つ目は、AI カーが白線認識をしている際、コースの真ん中を外れずに走行する事ができるかである。3つ目は、カーブするときコースアウトせず滑らかに走行できるかである。以上の3点に着目して評価する。

次に、障害物検出の精度に注目する。具体的には、障害物を AI カーの走行中に突然目の前に置いた際、AI カーは反応して停止するのか、という点に着目して評価する。

最後に、標識認識の精度に注目する。具体的に2つの観点から評価する。1つ目は、カスケード分類器は正常に標識を検出する事ができたのかである。2つ目は、標識を判定する CNN は止まれを判別する事ができたのかである。以上の2点に着目して評価する。

（文責: 青柳心吾）

## 第 4 章 結果

### 4.1 前期の結果

前期までの結果は 4 つある。主にテーマの選定、AI カーを作成する際に必要な前提知識や技術の習得、物品の購入を行った。

#### テーマの選定

テーマの選定を行った。私たちのグループの目的は既存の技術を応用することであった。そのため、脳の仕組みを利用した既存の技術をグループメンバーで調べ学習を行った。その結果、機械学習を利用した技術のテーマ案が話し合われた。具体的には、「自動運転」や「自動レジ」などである。自動運転については、ラジコンを用いて自作したコースを自走する事を目標とした。自動レジについては、公立はこだて未来大学の食堂に、カメラの画像からメニューを判別するレジを開発することを目標とした。多数決の結果、ラジコンに Raspberry Pi 4 を導入し、自作したコースを自走する。また、コースに標識を導入し標識に従った挙動をすることを目標とした。

#### 技術の基礎学習

開発に用いる技術について基礎的な学習をした。まず、自動運転に使われる技術である画像認識については、各自で調べ学習を行った。その結果、画像認識を実装するプログラミング言語として Python が主流であることや、主に使われるライブラリについて知る事ができた。具体的には、画像データの前処理には OpenCV、行列計算など特徴の抽出に必要な処理に NumPy、画像の判断や分類など機械学習の実装に Keras などがよく使われていることが分かった。画像認識について学習したのち、開発に用いる Python 言語の基礎学習を行った。学習教材としては、本プロジェクトの担当教員である香取教授に用意していただいた「データサイエンス入門」の講義資料を用い、Python 言語の記述を身に付けていくとともに、OpenCV や NumPy といったライブラリにも触れていった。

#### 物品の購入

開発に用いる物品を購入しそれらの環境構築を行った。購入した物品は「Raspberry Pi4」、「Picar-X」、「キーボード」、「マウス」、「Raspberry Pi4 の参考書」である。AI カーに搭載するコンピュータには、Raspberry Pi4 を選んだ。Raspberry Pi4 の初期設定を行い、環境構築を行なった。その後、Raspberry Pi4 を組み立て、カメラモジュールを取り付けたうえで、Python やライブラリのインストールなど、開発を進めるための環境を整えることができた。そして画像認識の基礎を行うために、カメラモジュールの映像からリアルタイムでの顔認証を実験として実装した。

#### 「Picar-X」の組み立て

購入した「Picar-X」の仮組み立てを行った。私たちのグループでは、ラジコンカーなどの組み立てを行ったことのあるメンバーがいなかった。そのため、まずは制作する AI カーについて全体を把握するために、私たちの成果物に近い「Picar-X」を購入した。組み立てをしていく中で、ラジコンの操作を行うにはステアリングやモーターを調節することが必要であると分かった。また、「Picar-X」は白線を自動で認識し、白線の内側を走ることができるため、そのプログラミングを行なった。

## 4.2 後期の結果

後期までの結果は3つある。主に AI カーを完成させるために、白線認識、障害物検出、標識認識の機能の開発を行った。

(文責: 青柳心吾)

### 白線認識

白線認識では主に3つの技術を使用した。1つ目は白線検出で、カメラで取得した画像から白線を検出し、その中心と AI カーとのズレを求めた。2つ目は PID 制御で、白線検出で求めたズレの値を元にステアリングを制御した。3つ目は PWM 制御で、カーブの時のなどのスピードを制御した。

白線検出について、今回我々は白線の色のみを利用した白線検出を実装した。機械学習を使うことでより正確に白線を抽出する方法も検討したが、技術的、時間的問題から実装することはできなかった。白線検出に用いるカメラについては、初めは一般的な Raspberry Pi4 用のカメラモジュールを使用した。画角が狭いため2本の白線とその外側にある標識までを一度に捉える事ができなかった。そこで、Raspberry Pi4 用の広角レンズが付いたカメラモジュールを使用する事でこれを改善する事ができた。しかし、広角レンズを用いた場合においても、AI カーがカーブの端にいるなどの特定の条件下では白線を2本同時に捉える事ができなかった。白線検出の精度には、白色抽出、2値化処理、逆透視投影変換が大きく関わっており、この中のいずれかの調整が悪いと白線検出の精度が大きく落ちた。白色抽出については、白色の中でも完全な白色 (RGB:255,255,255) を除くことで、太陽や照明などの反射光を誤検出することが減り、検出精度が大きく向上した。また、白色抽出の精度が白線検出の精度に最も大きく関わっていた。2値化処理については、閾値を小刻みに変えてその都度白線の抽出具合を確認し、最も白線だけを正確に抽出することができる閾値を模索することで検出精度が向上した。逆透視投影変換については、カメラの地面に対する角度によって正しい変換倍率が変わるため、AI カーのカメラの角度に合わせて調整を行った。変換倍率は、2本の平行な直線となっているコースを AI カーから撮影し、変換後の直線が水平になっているかどうかで決定した。今回は、画像の上辺の長さが下辺の6倍になるように画像を引き伸ばすと、丁度上から見下ろした形に変換することができた。また、白線の検出では画像のピクセルを1つ1つ調べることで処理を行っていたため、CPU への負荷が大きく、我々が使用した Raspberry Pi4 の性能ではカメラから取得した画像をそのまま使用して処理を行うことが困難であった。そのため、画像のピクセル数を約10分の1まで落とすことでリアルタイムでの処理が可能となった。最終的な白線検出の精度としては、コースの直線部分においてはほぼミスなく白線を検出し、その中心を求めることが出来た。しかし、カーブ部分では、AI カーの位置によって白線の片側がカメラの視覚から外れる場合があり、そのような状況下では白線を正確に検出することが出来なかった。

PID 制御については、当初は比例動作、積分動作、微分動作の3つを取り入れた完全な PID 制御の使用を検討したが、時間的問題から実装することは出来なかった。その代わりとして、P 制御と PI 制御の2種類を用いて AI カーのステアリング制御を行った。P 制御では、AI カーと白線中央とのズレの値に比例ゲインをかけた物を微分動作として、AI カーが白線の中心を滑らかに走行するように比例ゲインの値を調整した。しかし、比例ゲイン1つでは直線部分とカーブ部分のどちらか片方でしか白線の中心を走らせることが出来なかったため、比例ゲインの値を直線部分とカー

ブ部分で変更することで対応した。結果としては、P 制御を用いて円形のコースを自走して周回することが出来た。PI 制御では、AI カーと白線中央とのズレの累積値に積分ゲインをかけた物を積分動作として、積分ゲインの値を調整をした。しかし、累積値を用いるためカーブが続くと極端に値が大きくなってしまい、内側に曲がりすぎてしまった。そのために積分ゲインの調整が難しく、比例ゲイン、積分ゲイン、積分時間の3つの値を調整する必要もあったため、調整が難航した。結果としては、PI 制御を用いた場合では円形コースの周回はできず、直線のコースを走るという限られた状況の中でのみ白線の間を走行することが出来た。

PWM 制御については、ステアリングには角度の限界があり、PID 制御だけではカーブを曲がりきれない事があったため、そのような場合に PWM 制御を使ってカーブ時の速度を落とした。しかし、AI カー本体の性能の問題により PWM 制御だけでは我々が望む速度までスピードを落とすことが出来なかったため、プログラム側でモーターの ONOFF を繰り返すことで対応した。

最終的な白線認識の結果としては、P 制御を用いた場合でカメラが白線を2本捉えられている時に限り、コースを周回することができた。しかし、AI カーのスタート位置によっては白線をカメラで2本同時に捉えることが出来ない事があり、そのような場合にはコースを周回することが出来なかった。

(文責: 芥川裕大)

### 障害物検出

障害物検出には超音波センサーを使用した。AI カーの前方に超音波センサーを搭載しコース内の障害物を検出すると停止するように設定した。AI カーと障害物との距離が 20cm から 15cm で停止することを想定し作成した。そのため、プログラム上では、AI カーから 20cm 以内に障害物を検知したときに停止し、検知しなければ走り続けるような動きを作成した。実際に障害物の検出実験を行ったところ、カーブ上での障害物検出は、AI カーのスピードが最低速度になるため、想定していた距離で停止することができた。しかし、コース内の直線上に障害物を設置した場合は、AI カーのスピードが最高速度になってしまいタイヤを急に停止することが出来ず、想定していた距離で止まらなかった。何回か物体検出の走行実験をしたところ、直線上に障害物がある場合、8割が走行中に停止することができ、2割が衝突しまった。また、8割の停止できた間の距離は 10cm 未満となってしまった。

(文責: 中嶋彪吾)

### 標識認識

標識認識には主に2つの技術が必要であった。1つ目が標識の検出である。これにはカスケード分類器を使用し、車体に搭載されたカメラから標識の領域を検出した。2つ目が標識の判定である。これには CNN モデルを使用し、標識が「止まれ」であるかの判定を行った。

1つ目の標識の検出に使用したカスケード分類器は、学習画像の枚数を変化することで精度の向上を図った。特にポジティブ画像とネガティブ画像の枚数の比率を重視した。まず初めに、ポジティブ画像を 2,000 枚に固定し、ネガティブ画像の枚数を変更することでより精度が良い比率を明確にした。比率を 1:0.5、1:1、1:2 としてカスケード分類器を作成した。その結果3つの中で一番精度が良かったのは 1:0.5 であった。しかしまだ十分な精度ではなかったため、1:0.5 をもとに改善を行った。それにより、7:3 と 6:4 の比率の精度が良かったため修正し 6.5:3 の比率が最も精度が良いという結果となった。さらにポジティブ画像を 6000 枚に変更し、6.5:3 の比率でカスケード分類器を作成したところ、ポジティブ画像が 2000 枚の時よりも精度が向上した。以上の結果か

## Make Brain Project

ら、ポジティブ画像を 6000 枚、ネガティブ画像を 2770 枚でカスケード分類器を作成し、実際に AI カーに搭載した。そして AI カーに搭載されたカメラからの情報で実験したところ、走行中に標識を検出することに成功した。

次に、CNN モデルを使用した標識判定では、作成したモデルを実際に Raspberry Pi4 に導入し検証しようとしたが、モデルに必要なモジュールを Raspberry Pi4 にインポートできなかった為、検証する事ができなかった。そのためモデルの検証は自分の PC で行い標識を判定した。以下に標識判定を行っている図 4.1 を示す。



図 4.1 CNN モデルを使用した標識判定

(文責: 青柳心吾)



## 第 5 章 考察

### 5.1 前期

前期に関して良かった点は、まず開発に用いる環境構築や Python 言語のような技術に関する学習をグループ全員がプロジェクト時間外でも個人で行い、互いに知識共有を行うことで知見を深めた点である。また、担当教員である香取教授にも協力をしてもらい「データサイエンス学習」の講義資料によって、様々なライブラリにも触れた。これにより、今後の活動がスムーズになり、AI カーの機能の調査や作成などで何を用いる可能性があるのか、用いた場合に使えるライブラリは何かがあるかなどを考える際に、円滑に行動を行うことができた。次に、早い段階で AI カーの仮組み立てを行った点である。前述したとおり、グループメンバー全員がラジコンカーなどの組み立てを行ったことがある者がおらず、そのまま作成に入ってしまうと目的のものがしっかりと作成できない可能性があった。また、最終成果物の性能を考えた際に、どのような部品を使えば想定している機能を実装できるのか、どのような配置であれば機能や動作の邪魔にならない配置ができるかなどの考察のためでもあった。この時に仮組み立てで成果物に近いものを作成したことにより、最終成果物の組み立てにおいて大いに役に立った。

前期における課題点は、まず全員がほとんど同じことを行っていたため、活動をスムーズに行うことができたが進みが遅かった点である。オンラインでの活動においても、対面での活動においてもグループメンバーのほとんどの人が同じことをしていたためにオンラインで出来ることを対面でやっつけてしまっていて、オンラインでのプロジェクト学習の時間が丸々一日分無駄になってしまうことで進みが遅れて、コロナウィルスによって数が少なかった対面での利点をほとんど活かすことができなかった。次に、AI カーの仮組み立てに時間を割きすぎてしまった点である。そもそも、前述したとおり進みが遅かったことに加えて仮組み立てに想定以上に時間をかけてしまった。また、この時に AI カーの仮組み立てに必要な以上に人数を割いてしまったので、最終成果物に用いる機能の調査や Python 言語の勉強等の時間を減らしてしまった。これにより、最終成果物の考案や必要となる物品の注文・組み立てのように最終成果物に直接関係する本組み立てへの着手が遅れてしまった。

これらの課題点について、一つ目の課題点に対しては AI カーに用いるラジコンの組み立てや様々な設定等を調査、調節を行うラジコン組と最終成果物に搭載する機能の実装方法の調査、プログラムの作成等を行う機械学習組の 2 つのグループに分けることで活動を今よりも迅速に行えるようにして、進みの遅さの改善及び、専門的にすることで必要となる技術の調査の時間の低減を行った。二つ目の課題点に対しては、夏季休暇中に本組み立てに用いる物品をあらかじめ調査を行い、後期でのプロジェクト学習が始まれば即座に注文を行ったり、プロジェクト学習の時間外でもそれぞれの組で連絡等を行い、後期に入れば即座に本組み立てや最終成果物の作成・機能の作成等に着手できるようにした。

(文責: 坂中武蔵)

## 5.2 後期

後期に関して良かった点は、前期の課題点に対しての行動をしっかりと行えたことである。ラジコン組、機械学習組と分けたグループはしっかりとそれぞれの組で想定していた活動を行うことが出来た。また、本組み立てに関しても後期のプロジェクトの活動が始まると同時に最終成果物に直接関係する組み立てや機能の作成等に着手することが出来、搭載する機能の詳細や機能に必要な物品の吟味を時間をかけて行うことが出来た。また、当初の目的であった白線を認識してはみ出さないように走る機能・障害物を認識して自動で停止を行う機能・標識を認識して標識の指示に従う機能の作成・搭載を行い、最終成果物を自作のコースで自走させるということを行うことが出来た。次に、組内での活動の分業をしっかりと行っていた点である。前期での失敗を糧として、それぞれの組で全員がずっと同じことをやり続けることや、その時でなければできない活動をやらない事などが起きないように活動を分けていた。これにより、前期のような進みが遅くなり重要なことができてない場合がなくなった。

後期における課題点は、グループを分けたことによって互いの進捗や機能のかみ合い等をしっかりと把握できなくなってしまった点である。片方の組で、使用を考えていた機能の方法が搭載することが不可能であり他の機能の搭載方法を最初から考え直さざるを得なくなったり、そもそも成果物の本体が存在しないと作成が困難なものであるため対面等の本体に触れる機会がなければ作成を行うことが出来ないため他のことをやらざるを得なくなったり、その日のプロジェクト活動を終えるしかなくなるなどの要因が相次いだために、度々活動を止めざるを得なくなってしまった。また、他の組が最初から考えていた機能を良かれと思って作成・改良等を行ってしまったせいでグループメンバー内で意見の衝突や考えの食い違いが起きてしまう事もあった。次に、それぞれの班の中のメンバーで活動の重さに差ができてしまった点である。例を挙げるとすると、ラジコン組でのラジコン本体の組み立てとステアリングやスピードメーターの値の確認では、前者の組み立てが説明書のような解説があるのに対して、後者は正解がその時々で変わってしまうため、後者の活動の方が重くなってしまい、責任やプレッシャー等がその人に集中してしまうという事である。

これらの課題点については、まず一つ目の課題点については、互いに使用を考えている技術について、グループ内のメンバーにしっかりと相談を行い、可能であるか、他の機能の妨げにならないかなどを考えて搭載を行うものとしたり、組に関係なしにどんな時でもコミュニケーションを忘れないように心掛けを行った。また、どんな小さなことでも質問ができるような雰囲気グループメンバー一人一人が心がけた。二つ目の課題点については、組内で軽い活動だけを行う人であったり、重い活動ばかりを行っている人を出さないようにそれぞれの組で重い活動であれば複数人で、軽い活動であれば手の空いた人がやるといった風に、組の中で一人だけに責任やプレッシャー等が集中してしまう事が無いように努めた。

(文責: 坂中武蔵)

## 第 6 章 外部評価

### 6.1 中間発表

#### 中間発表の評価

中間発表は 2021 年 7 月 9 日に開催された。今年度も新型コロナウイルスの影響により、Zoom を用いてオンライン形式で行われた。発表方法は紹介動画を撮影するか、Web ページを作成するかの選択であった。我々のグループでは、紹介動画を選択した。発表用のスライドを作成し、そのスライドに沿った発表音声を収録した。聴講者は各グループによる発表の視聴後、それぞれのブレイクアウトルームで行われる質疑応答に参加した。質疑応答は前半と後半があり、1 回 15 分をそれぞれ 3 回ずつ、聴講者を入れ替えて行われた。そして、事前に作成していたアンケートフォームに評価を記入してもらった。評価を行う項目は発表技術についての評価と発表内容についての評価の 2 項目についてである。それぞれの項目について 1 点から 10 点の点数での評価をしてもらい、評価の理由やアドバイスなどをコメントとして記入してもらった。中間発表で得られた点数による評価を表 6.1 に示す。また、質疑応答とコメントの一部を以下に示す。

#### 質疑応答

- ・環境構築はうまく行ったのか  
→うまくいきました
- ・AI カーに使用するラジコンはどのようなものを使用するのか  
→市販のものを使用する予定です
- ・Raspberry Pi4 を使用するのなぜか  
→Python 言語と相性がよく、PWM 制御を用いて駆動系の操作を行う事ができるため
- ・自分達が決めたコース以外を走らせることは想定しているのか  
→想定してない
- ・現在の自動運転の車が事故を起こした際の責任問題についてはどう考えているのか  
→責任問題については考えていない
- ・ラジコンを用いた自動運転で実際の事故を減らすようにはできるのか  
→自動運転に使用する技術の理解を重視していたため、事故を減らすことは考えていなかった

(文責: 青柳心吾)

#### 発表技術

- ・イラストや画像を用いており内容が把握しやすかった
- ・動画もポスターも丁寧に作成されていて見やすかった
- ・「AI カー」のスライドや音声の進行が速すぎると感じた
- ・スライドを用いた説明で視覚的にも分かりやすかった

## 発表内容

- ・スライドをみせるのがわかりやすいと思いました
- ・目的はあるが、目的から何を得られるかが気になる
- ・質疑応答がちゃんと答えられていてよかったですと思います
- ・具体的にどのように組み立て実験していくのかわからなかった

表 6.1 中間発表

評価	発表技術	発表内容
10	2	3
9	6	6
8	10	9
7	8	6
6	6	9
5	4	2
4	0	1
3	0	0
2	0	0
1	0	0
合計	36	36
平均	7.4	7.4

## 評価の考察

点数による評価を振り返ると、表 6.1 より発表技術の平均点は 7.4 点、発表内容の平均点は 7.4 点であった。発表技術においても、発表内容においても、1~3 点といった極めて低い評価は見受けられなかったところは良かったと感じる。しかし、9 点や 10 点といった高い評価が秀でて多いわけではなく、中程度の評価の点数に一番多く人数が配分されていたので、後期の成果発表ではより良い発表を目指したい。聴講者から得られたコメントを振り返ると、発表技術については、内容が把握しやすかったりスライドが見やすかったりといった良い意見を得られた。このことから、発表方法についてはスライドに沿って説明をしていくという形式でプロジェクトについて理解してもらうことができると分かった。そのため、発表方法については後期の成果発表においてもこの方法を採用したいと感じた。スライドについても、初めて内容に触れる聴講者にとっても分かりやすいように、良い意見を受けられた今回のものを参考にしたい。しかし、発表をしている音声が聞きにくいといった意見もあった。中間発表のための準備をする際、時間に追われてしまうことが多く、発表の収録に十分に時間をかけられなかったことが反省点として挙げられる。後期は成果物の制作を進めていくので、成果発表のための準備を行う際は、今回より時間に追われてしまう可能性も考えられる。そのような事態に陥らないためにも、前期の活動を振り返り、一つ一つの活動内容により見通しをもって活動を進めてくべきであると感じた。そうすることによって、発表資料の作成や収録にも十分な時間を確保したい。発表内容については、最終的な目標がよくわからないといった意見が多く見受けられた。前期の活動では、新型コロナウイルスの影響によってオンラインでの活動がメインであったこともあり、自動運転に関する基礎的な知識の習得や、Python 言語の学習、Raspberry Pi4 などの成果物の制作に必要な環境構築を進めることが主な活動であった。そのため、最終的な目標を具体的に定めることができていなかった。後期に着手する成果物の制作を計画

的に進めるためにも、目標を明確に決めていく必要があると考えられる。前期の活動を通して得られた情報や知識を元にグループ内で入念に話し合い、成果発表に向けて必要な活動を整理していきたい。発表技術と発表内容のどちらの項目においても、自分たちではなく外部からの評価を受けられたことによって、違った視点から良い部分や改善していくべき部分を見つけられた。また、これからの活動に活かしていける具体的なアドバイスももらったので、後期からの活動に有効に活用していきたい。

(文責: 折戸啓吾)

## 6.2 成果発表

### 成果発表の評価

成果発表は2021年12月10日に開催された。前期と同様に新型コロナウイルスの影響により、Zoomを用いてオンライン形式で行われた。発表方法についても中間発表と同様に、紹介動画を撮影するか、Webページを作成するかの選択であったため、我々のグループでは紹介動画を選択し、発表用のスライドの作成や音声の収録を行った。聴講者による評価も前期と同様に行い、聴講者は各グループによる発表の視聴後、それぞれのブレイクアウトルームで行われる質疑応答に参加した。質疑応答は前半と後半があり、1回15分をそれぞれ3回ずつ、聴講者を入れ替えて行われた。そして、事前に作成していたアンケートフォームに評価を記入してもらった。評価を行う項目は発表技術についての評価と発表内容についての評価の2項目についてである。それぞれの項目について1点から10点の点数での評価をしてもらい、評価の理由やアドバイスなどをコメントとして記入してもらった。成果発表で得られた点数による評価を表6.2に示す。また、質疑応答とコメントの一部を以下に示す。

### 質疑応答

- ・ AI 車同士で譲り合いなどについてはどのように考えているのか  
→複数台での走行は考えていなかった
- ・ 完璧に白線認識を行えているのか  
→些細な環境の変化でカメラFPSが変わってしまうため、完璧にできているとは言えない
- ・ 白線認識にセンサーを使用すれば精度が向上するのではないか  
→検討します
- ・ 透明な物体でも障害物検出を行うことはできるのか  
→超音波センサーを使用しているため、透明な物体でも検出は可能である
- ・ カスケード分類器に使用した画像の比率で6.5:3はなぜそうなったのか  
→7:3と6:4がうまくいったため改善を行った結果そのようになった
- ・ 走行する地面に凹凸があっても走行することはできるのか  
→PID制御のD制御が必要になってくるため、現状はできない可能性がある
- ・ 事故を減らすことはできたのか  
→自動運転に使用する技術の理解やその開発を行っていたため減らすことはできていない
- ・ 対向車や十字路での事故は減らせるのか  
→その想定は行なっていなかったため減らせてはいない
- ・ AI 車同士での衝突などの実験は行なったのか

## Make Brain Project

→行っていない

・Jetson を用いると処理速度はどのくらい変わるのか

→実際に使用はしていないため分からない

(文責: 青柳心吾)

### 発表技術

- ・図や写真や動画が適切に使われていて全体的にわかりやすい発表だったと思う
- ・動画がとても聞きやすくポスターも初見の人に分かりやすいよう簡潔に設定されていた
- ・文字ばかりのスライドや難しい説明が多いと感じた
- ・専門用語が非常に多く、解説が無かったためわかりにくかった

### 発表内容

- ・様々な機能を有していて面白いと感じた
- ・ポスターや動画などを通じて自動運転の仕組みについて理解することができた
- ・既存の技術との差異を見つけて掘り下げるのも面白いのではないかと思った
- ・何について理解することができ、それをどうするのかなどを伝えてほしい

表 6.2 成果発表

評価	発表技術	発表内容
10	5	4
9	8	12
8	10	13
7	7	5
6	2	2
5	4	0
4	1	1
3	0	0
2	0	0
1	0	0
合計	37	37
平均	7.8	8.1

### 評価の考察

点数による評価を振り返ると、表 6.2 より発表技術の平均点は 7.8 点、発表内容の平均点は 8.1 点であった。中間発表の評価と比較すると、発表技術は 0.4 点上がり、発表内容は 0.7 点上がっており、どちらの項目においても評価を伸ばすことができた。各点数における人数の配分を見ると、中間発表と同様に 1~3 点といった極めて低い点数は無かったところが良かった。それに加えて、9 点や 10 点といった高い点数の評価が中間発表よりも多く得られた。発表方法は前期と同様に動画内でスライドを用いて発表をする形式で行ったが、発表用の原稿の作成を、担当した作業を基準にしてグループ内で分担して行ったことによって、前期よりも時間に余裕をもって準備を進めることができたと思う。準備のための時間を確保することは、中間発表後の反省点の一つとして挙げられていたことであったので、その点においては、前期の反省を十分に活かし改善することができた

## Make Brain Project

と思う。そして、そのことがどちらの評価項目においても点数による評価の平均点を伸ばせられたことにつながったと考えられる。次に聴講者から得られたコメントを振り返ると、発表技術については、分かりやすいといった意見も得られた一方で、専門用語が分からず内容が十分に理解できないといった意見もあった。我々は一年間の活動を通して、発表内容について前提知識があるが、聴講者にとっては初めて触れる内容である可能性もある。資料作成を進めていくにあたって、聴講者の立場で考える意識をもっと持つべきであったと考える。また、スライドの作成や発表音声の収録は、担当したグループメンバーにばかり頼ってしまい、グループ内全体で完成したものの確認や改善案を出し合うことなどはできていなかった。原稿作成の分担に加えて、資料の作成や確認作業においても分担して行うことができているならば、より良い発表ができたかもしれない。資料を作成するための時間を確保するだけでなく、内容についても前期より入念に仕上げていくことができているならば、聴講者への配慮も十分にできていたのではないかと考えられる。発表内容については、良い意見として、成果物が面白いといった意見や活動内容を理解することができたといったものが得られた。成果物の目標設定が前期のものから変更となるが多かったが、学んだことを活用した作品を最終的に完成させることができ、このような良い意見を得られたことはとても良かったと感じられる。しかし、最終的にどのような成果を得られたのかといった意見や、既存の技術との差異はあるかといった意見もあった。成果物を完成させることを目標としていたこともあり、そこから実際の自動運転との差異を見出すことや、我々の活動内容をどのように実社会で活かすことができるのかといったことまで考えることができていなかったため、反省していく必要があると感じた。しかし、良い意見が多かったことから、発表を通して活動内容を理解していただけたのではないかと考える。

(文責: 折戸啓吾)

## 第7章 まとめ

### 7.1 前期、後期の取り組みについて

前期の活動としては、授業の3回に2回はオンラインでの授業であり、オンラインでの授業の際は、自動運転の仕組みや画像認識に関する論文を模索してその輪読を行うことが多かった。対面での授業の際は、使用する Raspberry Pi4 の環境構築や、Python 言語の基礎的な内容の学習、画像認識などの理解を深めるための実装などに重点を置いた。具体的には、香取教授から準備していただいた「データサイエンス入門」という科目の資料をもとに、メンバー全員が基礎的な演習を毎週行い、Python 言語に対する理解の助けとした。加えて、Raspberry Pi4 を用いて自動運転をするためにあらかじめ設計された AI カー、「Picar-X」を購入し、これを AI カーの一代目とした。それを組み立て実際に動かすことで、自分たちが作ろうとしている AI カーの仕組みに対する理解の助けとした。中間発表会ではこれを組み立てて得た AI カーの仕組みへの知見であったり、画像認識について得た知識を用いて AI カーにどのようにして組み込んでいくのか、という内容について発表した。

良かった点としては、プロジェクト学習の時間内外問わず、各メンバーが積極的に論文を探したり使用する技術に関する学習を進め、その知識の共有を行うことで知見を深められたということである。これにより、AI カーを開発する上で必要な技術であったり、搭載する技術の現実性などについて、早い段階から学んでいくことができたと感じている。加えて、一代目の AI カーである、「Picar-X」の作成に着手したことである。私たちの中には、ラジコンカーなどの工作経験がある人物がおらず、後期にそのまま作成に着手していたら AI カーの仕組みなどについて理解ができませんまま進んでしまい、目標としている成果物を作成できない可能性があった。そこで、仮組み立てとして成果物に近い AI カーを一度組み立てておくことで、後期行う AI カーの本組み立てにスムーズに導入することができた。

反省点としては、各人に対して明確な役割分担を行っていなかったため、プロジェクト学習の授業時間でメンバー全員が具体的に何をしたらよいか分からず、時間を無駄にしてしまうことがあった点が挙げられる。プロジェクト学習の授業時間はオンラインでの作業になることが多く、お互いが何をしているかについて思うような情報共有ができなかったため、このような反省点が出てしまったと考える。次週までに行うことを明確にしたうえで、誰がどの作業を行うか、というところまでをはっきりとさせる必要があった。

9月の下旬、夏休みの一部を活用し、週に二回、グループで集まる時間を設けた。そこでは、後期に残っている時間から逆算し、AI カーに搭載する機能を検討・決定した。それを受け、後期の円滑な作業進行に向けて、AI カーに搭載する機能の実現に必要な、AI カー本体や部品などについても模索し、夏休み中にそれらの注文を行った。加えて、メンバー全員を「ラジコン組」及び「機械学習組」のいずれかに役割分担し、そこで作業をそれぞれ進めていくことに決定した。



後期は、役割分担を受け「ラジコン組」と「機械学習組」でそれぞれ並行して作業を行った。ラジコン組は購入した二代目の AI カーの組み立て、白線認識や物体認識機能の搭載、それらの機能を搭載した AI カーの制御を主に行った。機械学習組は標識認識機能の搭載に当たって、カスケード分類器を作成しその改善を行っていた。加えて、標識認識の精度向上のために CNN モデルを用いた学習を行い、カスケード分類器と併せて精度の向上を試みたが、Raspberry Pi4 で学習モデルを用いた標識の判別は検証できなかった。モデルに必要なモジュールを Raspberry Pi4 にインポートできなかったためである。しかし、自分のノート PC 上で標識を判定することには成功した。

良かった点は、作業を分けることをしていなかったため、複数人が同じ作業を行ってしまったり、そのプロジェクト学習の授業時間中に何をやっていいかわからず時間を無駄にしてしまうケースが多かった、という前期の反省を生かし、グループ分けを行うことで役割分担を明確にすることができたことである。ラジコン組は AI カー本体の作成や制御を、機械学習組は AI カーに搭載する機能を、といった具合で行う作業を分けることにより、作業に無駄が少なくなり、グループ全体の作業を行う速度が前期に比べて大幅に増した。加えて、夏休み中に AI カーを作成するのに必要な部品、本体について調査し、それらの注文を完了させていたため、後期に入ってプロジェクト学習の授業が始まるのと同時に成果物の作成を開始することができ、スムーズな進行を実現することができた。以上の二点が、当初目標としていた白線認識、障害物認識、標識認識機能の実装、搭載をするにあたって大きく役立ったと感じている。

反省点としては、ラジコン組と機械学習組がそれぞれ独立して作業を行うことが多く、お互いの進行度合いを今一つ把握しきれないことがあったため、最後のラジコン組と機械学習組でやってきたことの合体がうまくいかなかったことが挙げられる。最後お互いが開発してきた成果物を合体させるときになってわからないことが次々に出てきてしまい、情報共有を行うことの大事さを痛感した。プロジェクト学習の授業内でそれぞれの班が簡潔に発表して終わるだけではなく、お互いの班で着手している内容について、わからないことを質問する場を自分たちで設けるべきであったと感じた。加えて、後期になって時間不足を感じる場面が多くあった。グループ分けを行ったことによって作業の効率は大幅に向上したとはいえ、機械学習組が作成していたカスケード分類器の改善や、白線認識機能を搭載した AI カーの制御が思うように進まなかったことも相まって、予定していたスケジュールのようにはいかないことが多かった。カスケード分類器の改善については、正解画像と不正解画像の比率はどのくらいにするのがいいのか、見つけ出すのに何度も検証を重ねたことで想定より時間を取ってしまった。白線認識機能を搭載した AI カーの制御に関しては、Raspberry Pi4 の FPS が、本体の温度や周りの気温の影響を受けやすく、走行させるたびに挙動が変わりコースの走り方に差が出てしまったため、正確にコースの中央を走らせることができるように何度も修正を行った。それにより、最後の最後までコードを書き換え続ける必要があり、想定よりも時間がかかってしまった。AI カーを作成するにあたって、直面する障壁についてあらかじめある程度予想し、そこにかかる時間を事前に準備しておくことが必要であったと感じている。

(文責: 近藤裕貴)

## 7.2 成果

前期はまずテーマの選定を行った。最初は「自動運転可能な AI カーの開発」のほかに、「必ず読み勝てるじゃんけんマシンの開発」、「食堂のトレイの上ののったメニューの自動精算機能の開発」などが挙げられた。しかし、おおまかな難易度を考えたときに「自動運転可能な AI カー」は標識認識をさせる際に用いる機械学習であったり、車体の駆動系の操作などの難しさが予想され、そのテーマに対してあまり前向きに検討を行ってはいなかった。そこで、駆動系の操作を行うにあたって PWM 制御の可能性について香取教授からご教示いただき、Raspberry Pi4 を用いることで PWM 制御を AI カーの駆動系に利用することができるのではないか、という説明を頂いた。それを受け、グループメンバーで議論を重ねた結果、最終的に「自動運転可能な AI カーの開発」というテーマに決まった。テーマが決まってからは、AI カーを作成するにあたって使用する Raspberry Pi4 の環境構築や、Python 言語の基礎的な内容の学習、および自動運転に関する論文や画像認識にまつわる資料の収集・輪読を行った。Python 言語の学習、自動運転に搭載する画像認識の技術の調べものを進めていくうちに、私たちが目指す成果物を作成するには、画像の前処理に使用する OpenCV、行列の計算に使用する NumPy、画像の判断や分類、機械学習の実装に用いる Keras などといったライブラリが重要になってくることを理解した。それらを踏まえて、Raspberry Pi4 の環境構築を行ったり、「データサイエンス入門」という科目の演習を経て Python 言語に関する基礎的な学習を行った上で、画像認識の一環として Raspberry Pi4 上で顔認証をするプログラムの実装を行った。加えて、一代目の AI カーである「Picar-X」を購入し、その作成に着手した。私たちの中には今までに本格的な工作に取り組んだことのあるメンバーがほとんどいなかったことによる購入である。それによって、Raspberry Pi4 を車体に組み込む際の構造であったり、実際に動かすことで AI カーの走行に関する知見が得られた。そこから、後期への方針として、AI カー本体の制御には Raspberry Pi4、および Python 言語を使用すること、二台目の AI カーを購入し、それを一から組み立てていく過程で様々な機能を搭載していくことを決定した。

後期はラジコン組と機械学習組に分かれ、それぞれ作業を行った。

ラジコン組は AI カー本体の組み立て、白線認識を用いた制御に重点を置いて活動していた。組み立てを進めていくうちに、スピードコントローラ、およびステアリングサーボに関して、Raspberry Pi4 を用いて動かすためには PWM 制御を用いるが、そのままではピンが足りないことを知り、I2C 接続という方法を活用することに決めた。モータドライバを経由して I2C 接続することで、ピンを増やすだけでなく PWM の値を楽に調整することも可能になった。本体が走りまわることが想定すると、コンセントから Raspberry Pi4 への電源供給は困難であるということも分かり、モバイルバッテリーを用いることで走行中でも Raspberry Pi4 への電源供給を可能にした。AI カー本体を走らせる際、スピードコントローラおよびステアリングサーボそれぞれに対して PWM の値を決める必要があり、その最大値最小値を探る必要があった。そこで for 文を回して値を探った結果、スピードコントローラの PWM 値の最低値は 280、最高値は 356 であり、ステアリングサーボの PWM 値の最低値は可動域限界で 350、最高 450 ということを明らかにした。白線内を直進する際、そしてカーブを曲がる際、車体がレーン中央より右側にあるときはステアリングを左向きにし、車体が左側にあるときはステアリングを右向きにするという簡単な方法では、車体が斜め向きのまま中央に向かってしまい、結果としては中央線を蛇行しながら進んでしまうような挙動になることを明らかにした。そこで、PID 制御という手法を用いることで AI カー本体を滑らかに動かすことに成功した。今回は PID 制御のなかの P 制御を用いた。車体とレーン中央と

の距離に比例した値をステアリングの値にすることで、車体がレーン中央から離れているほどステアリングが中央側を向き、車体がレーン中央に近づくほどステアリングが正面を向くようになる。そうすることで当初想定していた、車体がレーン中央より右側にあるときはステアリングを左向きにし、車体が左側にあるときはステアリングを右向きにするという方法よりも高い精度で中央を走行させられるようになった。そこで、I 制御を取り入れた PI 制御にも着手した。しかし、PI 制御になると直線の走行はある程度うまくいくものの、ゲインの値が二つになり、それを足し算する計算になるため、値の調整が難しく、時間的な猶予のなさも相まって実装することはできなかった。D 制御に関しては車体が押されたりして突然コースアウトした時など、急な変化に対応できるようになるものの、今回はそのようなシチュエーションは想定していなかったこと、加えて、I 制御と同様に時間的な余裕がなかったため実装することはできなかった。AI カー本体を走行させる際、最初はコードを Run した後、Raspberry Pi4 とモバイルディスプレイ、および接続していたキーボードやマウスを切り離して走行させていた。しかし、Raspberry Pi4 は振動に弱く、走行させるためにそれらを切り離す際に少しでも揺らしてしまうとフリーズして動作しなくなってしまい、再起動してまた初めからやり直す必要があった。そこで、SSH 接続を使用することで私たちが使用している推奨機のノート PC からでも Raspberry Pi4 内のプログラムを書き換えたり、動作させたりすることを可能とした。それらを踏まえた上で、コースの白線を認識し、カーブでも直線でもおおむねその中央を走ることのできるプログラムの作成に成功した。具体的には、Raspberry Pi4 の FPS の内、12 フレームに 1 回、スピードコントローラの PWM の値を 356 とし、それ以外の 12 フレームに 11 回、スピードコントローラの PWM の値を 355 とした。しかし、Raspberry Pi4 の FPS がその時の周囲の気温であったり、本体の温度などに左右されやすい関係で、走行実験を行う度通るルートに微妙な誤差が生じてしまい、完璧に白線の中央を走ることのできるプログラムの作成には至らなかった。

機械学習組は標識検出に使用するカスケード分類器の作成と精度の向上、標識判定に使用する CNN モデルを用いた学習に尽力していた。標識検出を行うには、SSD や YOLO といった物体検出アルゴリズムを使う手法が有名であるが、今回使用する Raspberry Pi4 には GPU が搭載されていないことから、処理が軽いカスケード分類器を使用することで、Raspberry Pi4 上でも安定して動くようにした。また、標識判定には画像認識の手法である CNN モデルを使用することを決定した。カスケード分類器を作成するには学習画像であるポジティブ画像とネガティブ画像が必要であり、その比率と枚数が精度に大きく直結することになる。様々な比率で検証を重ねた結果、使用するポジティブ画像とネガティブ画像の比率が 6.5:3 の時に検出の精度が優れていることを明らかにした。また、ポジティブ画像とネガティブ画像の比率を 6.5:3 として、ポジティブ画像が 6000 枚と 2000 枚の時のカスケード分類器の精度を比較したところ、ポジティブ画像が 6000 枚の時の方が精度は高かった。以上のことから、ポジティブ画像を 6000 枚、ネガティブ画像を 2770 枚でカスケード分類器を作成し AI カーに取り入れた。その結果、「止まれ」の標識を AI カーがカメラで読み取った際、その標識の手前で停止させるプログラムを作成することに成功した。CNN モデルを用いた学習については、カスケード分類器から得た標識の画像を「止まれ」であるか否かを判定するのに使用した。標識は「止まれ」のみであったため、「止まれ」の標識を正解画像、「止まれ」以外の標識を不正解画像として、それぞれ 3960 枚用意した。データの 8 割をトレーニングデータ、残りの 2 割をテストデータとして、500 回の学習を行った。モデルの検証は自分のノート PC で行い標識を判定する事ができた。そして作成した CNN モデルを Raspberry Pi4 に導入しようとしたが、モデルに必要なモジュールを Raspberry Pi4 にインポートできなかった為、AI カーに搭載する事ができなかった。最終的に、超音波センサを用いた物体認識、カスケード分類器を用いた標

識認識、カメラモジュールによる白線認識をしつつ、それぞれに応じた動作をするAIカーの作成をすることに成功した。

(文責: 近藤裕貴)

## 7.3 改善点

### 自動運転との違い

今回作成した「視覚をもつAIカー」は白線を認識し直線やカーブの走行ができること、標識を認識しそれに応じた対応ができること、障害物を認識し停止することができることの3点を機能とした。プロジェクト開始時に目標としていた機能を持たすことはできたが、自動運転にはまだ程遠い。より自動運転に近づけるためには以下の3つを改善点としてあげる。1つ目はより多くの道路標識に対応できること。2つ目は交差点など白線の補助がなく右折、左折、直進の動きができること。3つ目は複数のAIカーを作成しリアルな交通状況で円滑に動かすことである。

### より多くの道路標識に対応

1つ目を改善点と挙げる理由として、「視覚をもつAIカー」で認識できる標識は止まれ標識のみであり、他の標識を認識することができないためである。実際の道路では止まれ標識以外にも様々な標識が存在し、その標識の指示に沿った運転をしなければならない。

### 交差点での対応

2つ目を改善点として挙げる理由として、視覚を持つAIカーは道路の両脇の白線を認識して白線の間を走行することはできるが、交差点では右折する際は右側に白線があり、左折する際は左側に白線があり、直進する際は白線がなくなるため走行ができなくなるためである。視覚を持つAIカーは道路の両脇の白線を認識し白線と白線の間を走行できる。要するに1本道のみ走行ができる状態である。この状態では交差点が存在する実際の道路では走行することができない。

### 複数のAIカーの操作

3つ目を改善点として挙げる理由として、複数の車両が走行していると譲り合う状況が発生するが、私たちのプロジェクトでは視覚を持つAIカー1つしか作成していないため譲り合う状況が発生することは想定していなかったためである。AIカーを複数台にした際に、衝突しないようにするためにもそれぞれの対応を考慮する必要がある。

### 改善方法

これらの改善点を改善するためには、第1に、Raspberry Pi4よりも高速に処理を行うことができるコンピュータが必要である。例えば、Jetson Nanoなどである。第2に、止まれ以外の道路標識を認識させるためにCNNを用いた機械学習が必要である。第3に、進路交譲を行うアルゴリズム [1] を作成することが必要である。Jetson Nanoは今回のプロジェクトで途中から使用を試みたが、環境構築に時間を取られうまく行かなかったため使用することができなかった。初めからJetson Nanoを使用していれば、より高度な物体検知や高速での処理ができたと考える。Raspberry Pi4では、白線認識、AIカーの制御、標識認識を同時に行っていた。そのため、カメラのFPSが低下したため高速でAIカーを動かすと処理ができなくなってしまった。止まれ以外の標識の認識は、標識を認識させるとともに、その標識にどのような意味がありAIカーがどのような判断を行うと良いかを定義しなければならない。AIカーを動かすためにはPWM制御を使用しているため、PWM制御と連動させることが必要である。複数の標識を認識した場合、どの標識

## Make Brain Project

が最優先なのかも考慮した上でプログラムを作成する必要がある。複数の AI カーを使用した場合の進路交譲の方法は、それぞれの AI カーが次にどのような行動を行いたいのか通信し合うこと [16] で可能になるのでは無いかと考える。さらに、右折、左折、直進といった行動のどれを優先するかを明確にすることでより円滑に進路交譲ができるようになるのではないか。しかし、交通状況はその時その時で変化するため臨機応変に対応する必要がある。視覚を持つ AI カーでは機械学習、画像認識、PWM 制御、PID 制御、センサーなどの技術を利用したが、道路交譲を行うにはこれらの技術だけでなく様々な技術や知識が必要となるだろう。また、衝突する場合のことも考えるとトロッコ問題についても対応しなければならない。情報処理技術の枠を超えた考え方が必要となってくる。様々な領域の考え方や技術を組み合わせることで問題を解決しなければならない。以上の3つの改善点を全て改善させることはとても難しい問題であるがより高度な自動運転に近づけるためには必要不可欠である。そのため、今後の改善点として改善していきたい。

(文責: 寺井遼大)

## 第 8 章 個人の取り組み

### 8.1 近藤裕貴

#### 前期

最初のテーマ決めの際、PWM 制御の可能性について調査し、それをメンバーに共有したうえで、「自動運転のできる AI カー」を開発したら面白いのではないかと、という意見を積極的に出した。テーマが決まってからは、主に使用する Raspberry Pi4 の環境構築を行ったり、Python 言語の基礎学習、搭載する様々な機能の検討・調査を行った。Raspberry Pi4、および画像認識などの機能については触れた経験がほとんどなく、事前知識がなかったため、論文を探して読んだり、わからない点はともに学習を進めていたグループメンバー同士で質問することで理解を深めた。Python 言語については、香取教授から頂いた「データサイエンス入門」という科目の資料を基に、簡単なプログラムの作成を行うことで徐々に仕組みを理解することができた。具体的には、NumPy や Matplotlib などの主要なライブラリを使用しプログラムの作成を行うことでそれらの使用方法を理解し、活用することができた。画像認識などに用いる Open CV についてはインターネットで資料を探してそこから学習したり、画像認識の学習の一環として簡単な顔認識機能の実装を行ったりすることで知見を深めていった。また、グループリーダーとして自分の意見を出しつつメンバーの意見をまとめ、そこから方向を決めていくことが多くあった。そのほかに、グループメンバーの作業の着手状況をそれぞれ把握し、そこから指示を出した。

7 月の中間発表では動画、およびその原稿の作成を行った。動画に使用するスライド、およびスライドに使用する画像などについては、他のグループメンバーに頼ることで時間的な余裕をもって動画の作成を行うことができた。中間発表当日では自グループへの質疑応答の対応を担当した。当日になって自分たちが想定していなかった質問がいくつも飛んできたため、回答に焦ってしまうことが多かったと感じている。加えて、学んできた内容で理解しきれない部分も多く、平静であれば答えられそうな質問でもところどころ回答に詰まってしまう場面を見せてしまった。自分だけで全て請け負おうとするのではなく、時にはグループメンバーを頼ることをするべきであったと反省している。

前期を通して、知識の習得がメインとなっていたため、実際に手を動かして学ぶことがあまり多いわけではなかった。加えて最終的に作る成果物本体の購入をしていなかったため、成果物をどのような形で手を加えて仕上げていくのか、という予想ができなかった。後期には手を動かしつつ、AI カーとなる本体を購入して実際に手を加えていくことでプログラミングであったり、AI カーの全体像に関する理解を深めていく必要があると感じた。

#### 後期

後期は「ラジコン組」と「機械学習組」に分かれてほぼ独立して作業を行っていたため、グループリーダーとしてお互いの作業の進捗状況を把握しつつ、適宜指示を出すことが多かった。私はラジコン組の方に所属していた。具体的な活動としては、成果物を作成するのに必要な物品の情報を収集し、その検討・決定を行ったり、AI カー本体の作成の補助、コースの作成、AI カーの制御を行っていた。

後期は「ラジコン組」と「機械学習組」に分かれてほぼ独立して作業を行っていたため、グルー

リーダーとしてお互いの作業の進捗状況を把握しつつ、適宜指示を出すことが多かった。私はラジコン組の方に所属していた。具体的な活動としては、成果物を作成するのに必要な物品の情報を収集し、その検討・決定を行ったり、AIカー本体の作成の補助、コースの作成、AIカーの制御を行っていた。AIカーの作成の補助について、前期に「Picar-X」の作成に着手したとはいえ、私たちのグループメンバーはほとんど今までに本格的なラジコンカーなどを組み立てた経験がある人がいなかったため、事前知識がほとんどなかった。そうした中で、Raspberry Pi4 とスピードコントローラ、ステアリングサーボとの接続方法などについて可能な限り調査し、その手法について共有した。AIカーの制御について、私たちのAIカーは直線、曲線の走行時にPID制御のうちのP制御を用いる手法を採用していた。P制御はAIカー本体が動作しているとき、角度調整を行う上で比例動作を行う必要がある。そこで、それを行う上で適した数式を考え、提案した。そうして走行可能になったAIカーに白線で作成したコースを走らせて、明らかになった改善点などを分析し、PWM値や比例ゲインに対する修正案を提案したりした。具体的には、Raspberry Pi4は気温の変化や本体の温度の高さなどによるFPSの変動が激しいため、同じコードを走らせてもAIカーは違う挙動をすることが多かった。そのため、その都度制御面での調整を行った形である。

12月の成果発表では中間発表と同じく動画の原稿、および動画の作成を行った。中間発表のスライドに使用した画像について、著作権の問題で引っかかってしまうことが多かったため、今回はその反省を生かしてスライドに使用する画像や動画などについては極力自分たちのカメラで撮影するようにした。動画に使用するスライド本体の作成についても、中間発表と同様にほかのメンバーに作成を頼ることで時間的な余裕をもって動画の作成を行うことができた。

成果発表会当日では中間発表会と同様、質疑応答の対応を主に行った。中間発表の時よりも基礎的な知識レベルが上がったことで、前期ほど質問に対する回答で詰まることは少なかった。加えて、前期で行った中間発表の時の反省点を生かし、自分の専門外であることは無理に答えようとせず、ほかのメンバーに頼ることで、質疑応答は全体を通して円滑に進めることができたと感じた。プロジェクト学習全体の活動を通して振り返ってみると、私はどちらかというとプロジェクト学習の時間内ではメンバー全体をまとめるという、グループリーダーとしての側面が強く、プロジェクトで行った学習に対してはそこまで深入りすることがなかったと感じている。しかし、自動運転の基本に関する部分、それに必要な技術の基礎を学ぶことができ、そしてメンバーの中心となってグループ全体をまとめていくことは、相応のスキルをきちんと要求されるし、何か物事を協力して行っていく上で重要であり必要不可欠であるということを知ることができた。この経験は必ず糧になると感じている。

(文責: 近藤裕貴)

## 8.2 坂中武蔵

### 前期

私は活動の初期は、まずテーマ決めである。テーマ決めにおいて私は意見として出てきたテーマに対して、プロジェクトのテーマとして適切であるかやそのテーマをさらに良くすることはできないかなどの意見を出し、グループメンバーの全員が意見を出しやすいように積極的に質問を行った。テーマ決めが終わってからは、環境の模索・調査を行った。当初はopenCVやPython、今回のプロジェクトで用いたRaspberry Pi4などについてほとんど知識を持っていなかったため分からなかった部分はすでに知識として持っているメンバーに質問する、自分で調べて学習するなどを

## Make Brain Project

して、作業を行った。また、担当教員である香取教授に頂いた資料を元に簡単なプログラムの作成を行い、仕組みの理解を深める学習も行った。学習・知識共有後は、Python で openCV を用いた簡単な画像や写真から決めたものを検出するプログラムを作成した。その後搭載する機能に対して用いることのできるライブラリの多さや、Raspberry Pi4 との相性などから Python での作成が決まったため、AI カーのサンプルとして Python を用いて動かす Picar-X の作成・解析を行った。この作成・解析において、私は本品の作成や機能の理解・解析・成果物への応用方法など様々なものを担当した。この際、インターネットや Picar-X の制作を行っている本社にメールを送る等を行い、Picar-X に対する理解を深めた。また、成果物に搭載する機能の話し合いにおいて、サンプルを作成・解析を行った観点から自分達の考えている AI カーのためにはどんな機能が必須であるのかや本当にその機能は必要であるかなどの意見を積極的に行った。

中間発表では、自分の担当する原稿内容の作成を行った。また、LaTeX による中間報告書の作成の方法や LaTeX の使い方等をグループメンバーで知っている人がいなかったため、LaTeX の勉強と LaTeX による中間報告書の書き込みを全て行った。また、スライドに用いる画像や文字の大きさ等が適正であるかなどの添削を行った。発表日当日には、質疑応答を担当した。あらかじめ、その時間の担当のグループメンバーと打ち合わせをしていたこともあり、質問に対して滞りなく、受け答えを行うことが出来た。中間発表後には、自分の担当していなかった時間に質疑応答をしていたグループメンバーとどのような質問が来たか、どのような返答を行ったか、自分たちの最終成果物の質を上げれる質問はあったかなどの話し合いを行った。

前期はコロナのために三度に一度しか対面での活動を行えなく、明確なやる事が決定されていない活動が多かったため、オンラインで活動できること、対面でしかできないことの振り分けが出来ていなかった。また、Picar-X の作成・解析等に時間をかけすぎて、最終成果物に使う本体のことを何も考えれていなかった。後期では、ほとんどのプロジェクト学習の時間を対面での活動で行えるため前期のようにならないようにグループメンバーのそれぞれで作業を分担して円滑に活動を行い、最終成果物の作成にすぐさま着手できるように努めたいと考えている。

### 後期

後期に入る前に、夏季休暇を用いて最終成果物に用いる外身のラジコンや作成に必要な部品の購入を行ったが、その部品等の吟味の際には私は前期でサンプルを作成した観点からどのような部品が必要となるのかやどのくらいの大きさや性能を持つラジコンを用いた方が外身として有効的に活用できるかなどの意見を出した。後期では、前期で学習したことや活動を元にラジコン組、機械学習組というグループ分けを行い、最終成果物を作成する作業に取り掛かった。その際、私はラジコン組に所属した。私は、このラジコン組で最終成果物に用いるラジコンの中身の作成、ステアリング、モーターの調整、コースの作成等を行った。この作業の具体的な内容としては、成果物に必要な部分や不必要な部分を機能やカメラの位置等から決めて、取り付けや取り外しを行ったり、機能の搭載に関して Raspberry Pi4 やカメラの関係から台座や壁等の作成を行ったり、モーターの最低値・最高値を探ったりなどをした。この際、ラジコン組の中でも作業の細分化を行ったため、円滑に活動を進めることができ、問題が起きた場合にもすぐさま対応を行うことが出来た。しかし、機械学習組と別れたことにより互いの進捗状況や今必要としている技術・物品の把握が前期に比べておろそかになってしまったことが懸念点であった。

期末成果発表では、質疑応答に加えて司会を担当した。前期で司会をしていた人のやっていたことをしっかりと覚えていなかったため、所々で進行について注意を受けてしまった。また、前期同様今回もあらかじめ打ち合わせを行い、どんな質問には誰が対応するなどを決めていたが、予想していなかった質問に対して、対応が遅れて明確な回答を行うことが出来なかった。特に、今回のグ



ループの成果物では、自動運転を把握・理解することを目的としていたため、成果物が複数あった場合にも搭載した機能がしっかりと反応するのかや、この成果物で本当に事故を減らすことは可能であるのか等の質問に対して的確な回答を行うことが出来なかった。中間発表会と同様に、期末成果発表会でも受けた質問に対する話し合いを行った。中間発表の時よりも、質問に対してより深く考えをめぐらせ話し合いを行うことが出来た。

プロジェクトの活動を通して、私はプログラミングのようなプロジェクトに即した活動も初期には行っていたが、プロジェクトの活動の中で、ほとんどは AI カーのサンプルの作成や最終成果物の本体の作成のような土台に近い活動しか行うことが出来なかった。しかし、ある程度ではあるが機械学習やプログラミングに対して知見を深めることが出来たし、集団で何か一つの目的に向かっていく際に必要なことや必要なスキル、コミュニケーションを学ぶことが出来た。この経験は今後、自分が成長していく上で必ず必要であった経験であるため、自身の今後に活かしていきたい。

(文責: 坂中武蔵)

## 8.3 寺井遼大

### 前期

私は主に Python 言語の学習、使用する技術・知識の検討と学習、Raspberry Pi4 の環境構築を行なった。Python 言語、画像処理、画像認識、Raspberry Pi4 に関する知識は全くなくそれぞれ 1 からの学習であった。

まずは、Python 言語を学習すると同時に画像処理技術と画像認識技術の学習を始めた。Python 言語の学習は、事前に香取教授から「データサイエンス入門」という講義の資料を基に、NumPy や Matplotlib などの主要なライブラリの使用方法を実際にプログラムを書くことで学習した。画像処理や画像認識に必要なライブラリである OpenCV の学習はインターネットや論文で調べて個人的に学習を行なった。簡単な画像認識として顔認識を実装した。カメラでリアルタイムに顔を認識させ四角で囲むというプログラムを作成した。OpenCV を使用し、すでに作成されている顔認識のカスケード分類器 (haarcascade\_frontalface\_alt.xml) を使用することで顔認証を実装した。その際に身につけた OpenCV で画像を処理する方法やカスケード分類器をダウンロードし使用する方法をグループ内に共有した。これらの技術を共有することによりグループ全員の知識を向上させることができた。視覚を持つ AI カーを作成する際に必要な技術・知識の検討と学習を行なった後、Raspberry Pi4 の環境構築を行なった。視覚を持つ AI カー作成のために Raspberry Pi4 を購入し環境構築を行なった。Python のバージョンアップや NumPy などの外部ライブラリのインストールとバージョンアップを行なった。開発環境には Jupyter Notebook を使用するため、Jupyter Notebook のインストールも行なったそれぞれのインストール終了後に Raspberry Pi4 にカメラモジュールを接続させサンプルプログラムとして顔認証のプログラムを実装した。しかし、Jupyter Notebook で実装するとカメラモジュールの停止が煩雑であったため、Jupyter Notebook ではなく標準エディタである nano を使用する方針とした。

中間発表会では、報告書の作成に携わった。前期はオンラインでの活動が多かったということもあり、知識を身につけることがメインであった。そのため、報告書の内容はどのような知識を利用して AI カーを制作するのかであった。そこで、私自身報告書を作成すると共に、報告書や発表スライド作成を円滑に進めるため身につけた知識や情報をメンバーに共有した。

前期は知識・技術を身につけることに時間を費やしていたため、実際に AI カーを作成するまで

に至らなかった。Raspberry Pi4 の環境構築はできたが、簡単なプログラムの作成しかできなかった。学習すると共に AI カーを作成するなどもっと効率が良い活動ができたと感じている。後期では技術のインプットとアウトプットを効率的に行い、AI カーの作成に着手したいと考えている。

### 後期

後期では AI カーの駆動系を開発する「ラジコン組」(4人)と、白線認識や標識認識などの認識するためのプログラムを開発する「機械学習組」(3人)の2つに分かれて活動を行なった。私は前期に画像処理や画像認識について学習していたため「機械学習組」に配属し活動を行なった。主な活動内容としては、白線認識の実装と標識認識の実装が挙げられる。後期の途中からは主に標識認識の実装に力を入れ活動した。

走行させる案として、道路の中央に線を引きセンサーを用いて走行させる方法と、センサーを用いずに白線を認識させ走行させる方法の2つを挙げた。それぞれの案について調べた結果、前者は実装が簡単であること、後者は自動運転技術で実際に利用されていることがわかった。私たちは自動運転に近い視覚を持つ AI カーを作成したいため、後者のセンサーを用いずに白線を認識させ走行させる方法を選択した。白線認識のプログラムでは ROI 変換、IPM 変換、2 値化を行なった。白線認識のプログラム作成した後は、視覚をもつ AI カーを真っ直ぐ走行させるためにはどの様にすれば良いかを検討した。インターネットや香取教授にどのような手法があるかを聞いた。その結果、カメラモジュールから送られる画面の中心と左右の白線の中心との差を求めることで真っ直ぐ走行させることができるのではないかと提案した。そして、画面の中心と左右の白線の中心との差を求めるプログラムの実装に取り組んだ。

標識認識を行う案として、YOLO や SSD などの物体検出を行う方法と、カスケード分類器と CNN を利用する方法の2つを挙げた。前者は GPU での処理が必要となり Raspberry Pi4 での使用は難しいため、後者の方法で実装した。カスケード分類器を作成したことがなかったため、インターネットや論文で学習しメンバーに共有した。まず、サンプルとして枚数の少ないカスケード分類器を作成し、作成できたことを確認後メンバーに具体的な作成方法を共有した。メンバーは3人いたため、それぞれ画像の枚数を変更しカスケード分類器の作成を行うように指示を出した。カスケード分類器の作成にはポジティブ画像(正解画像)とネガティブ画像(不正解画像)それぞれをベクトルファイルとしなければいけない。そこで、どのような画像を使用するかなども検討し、3人で効率よく作成できるように割り振りをこなした。カスケード分類器を作成するためには使用画像枚数により異なるが、多くの学習時間が必要である。また、より精度の良いカスケード分類器を作成するために手法を調べていると、ポジティブ画像を水増ししベクトルファイルを作成する方法にも様々あった。そのため、いかに3人で効率よく作成することができるかを考えながら活動を行なった。具体的には、まず作成して精度がどの様に変化するかの確認してからメンバーに共有し精度の向上が望めそうな場合のみ作成方法を変更した。認識した標識が何かを判別するために CNN を利用した。しかし、白線認識や精度の良いカスケード分類器の作成に多くの時間を割いてしまったため、深い知識を学習し利用することは難しかった。機械学習や深層学習についてインターネット、参考書、論文にて学習したが、カスケード分類器の作成に時間を取られたため CNN モデルの構築などはできず参考書に記載されているモデルを引用する形となった。

プロジェクト学習を通して、Python 言語、画像処理、画像認識、機械学習の技術だけでなく、開発テーマの設定や問題を解決するために必要である問題解決能力を身につけることができた。また、決められた期間内で成果物を作成することの大変さを実感した。私のグループは7人と人数が多く、人数が多い故に役割分担の重要性やコミュニケーションの大切さを身にしみて感じた。誰か1人が多く活動すれば良いのではなく、全員がモチベーションを高く持ち、その人その人に合った

作業を分担しなければならない。そのためには、誰かがどの様なことに長けているのかを分析しコミュニケーションを取り情報を共有しなければいけないと考える。プログラミングの知識だけでなくグループで成果物を作成すること難しさややりがいなど多くのことを学ことができた。この経験を卒業研究やその後の研究などに活かしてより良い研究を行いたい。

(文責: 寺井遼大)

## 8.4 中嶋彪吾

### 前期

私の前期の主な活動は、自動運転についての技術やシステムについて調べることと Python や OpenCV を学習することだった。機械学習では主に Python 言語が使われているという点から「データサイエンス入門」の授業資料を基に早い段階からグループ全体で勉強を進めることができた。しかし、どのような AI カーを制作するか定まった目標を立てていなかったため、各自自動運転について調べた情報を共有しあうことに時間を費やしてしまった。そのため、もう一度グループ全体でどのような AI カーを作るか話し合うことで、それに必要な知識や技術を絞って学習することができた。

中間発表では紹介動画に用いるスライドを作成した。スライド作成にあたって画像の著作権問題で紹介動画提出日ぎりぎりの完成となってしまった。後期では、著作権問題になることなく見やすいスライドを提出期限に余裕をもって作成させたいと思った。

夏休みは、週 2 回大学に集まって活動をした。前期のほとんどの時間は、誰がどんな作業をしているか分からず成果物の作成に取り組むことが出来なかった。そのため、「ラジコン組」と「機械学習組」に分かれることにした。そうすることで自分が AI カーを作成する上で何について学び作業すべきかが明らかになった。また、どんな AI カー (機能やシステム) を製作するか改めて目標を明確にし、活動をした。

### 後期

後期は「ラジコン組」と「機械学習組」に分かれて作業を行った。後期の活動も前期同様、3 回に 1 回しか大学に集まる事が出来なかった。そのため、ラジコン本体やモーターなど「ラジコン組」に必要な物品を持ち帰り家での作業やリーダーの家に集まっての活動となった。

私は、後期初期、AI カーのスピード制御やステアリング制御を行った。始め、Raspberry Pi4 でモーターやスピードコントローラーを動かす方法が分からなかった。そのため Raspberry Pi4 の GPIO ピンや PWM を制御するためのライブラリについて各自調べ、情報や知識を共有した。何回も試行錯誤をした結果、実際に、Raspberry Pi4 でスピード制御・ステアリング制御を同時に制御することが出来た。Raspberry Pi4 での制御が可能になってからは「機械学習組」と共同で作業を行った。後期中盤では、白線認識と PID 制御、AI カー制御のプログラムを用いてコースを走るように調整した。コースを正確に走るまでに結構な時間がかかったが試行錯誤や数値の調整で自走させることができた。

後期終盤では、AI カーの機能であるセンサによる障害物検出のプログラムを作成し、機械学習組と標識認識のためのカスケード分類器の作成、実験・検証を行った。本来、CNN を AI カーに搭載する予定だったが、Raspberry Pi4 の環境構築が成果発表会ぎりぎりになってできなかったため、実装することが出来なかった。

成果発表会では、前期の中間発表同様に紹介動画のスライドを作成した。後期は、前期の問題を

踏まえた上で、著作権に気を付け、余裕を持って制作することができた。成果発表会当日では、後半の質疑応答の対応と質問内容をメモすることを行った。中間発表の時に比べ、より深いところまで質問をされる可能性があったため、どんな質問をされても答えられるように準備した。そのため、質問には的確に答えることができたと感じた。

プロジェクト学習を通して、私は AI カーの制御や障害物検出、標識認識の開発に携わることができた。開発を進めることで、自動運転技術の知識や Python 言語のプログラミングスキルを習得することができたと思う。今回の活動を通して、集団で一つの目標・目的を達成させるためには、知識やプログラミングスキルは勿論のこと、メンバーとのコミュニケーションが特に重要だと感じた。この経験は、今後社会に出たときに必ず役に立つと感じている。

(文責: 中嶋彪吾)

## 8.5 芥川裕大

### 前期

前期の活動としては主に、製作物に必要な技術や材料の調査、Python や OpenCV、Raspberry Pi4 の環境構築とそれらの学習、後期に向けた役割分担を行った。

製作物に必要な技術や材料の調査では、製作物の作成にあたって必要になるであろう技術や材料などを調べて共有した。Python と OpenCV の環境構築では、他のメンバーと情報を共有しながらそれぞれのパソコンで Python と OpenCV が使用できるように環境構築を行った。また、Raspberry Pi4 の環境構築では、他のメンバーと協力しながら環境構築をし、Raspberry Pi4 で Python と OpenCV が使えるようにした。Python の学習のために「データサイエンス入門」の授業資料を香取教授から貸していただき、各週ごとに課題を提出しながら進めていった。また、OpenCV の学習はメンバー同士で学習資料を共有しながら進めていった。

夏休みにも活動日を設け、その中で後期に向けた役割分担をした。具体的には、AI カー本体の作成を担当する「ラジコン組」と、白線認識や標識認識のプログラムを作成する「機械学習組」の二つのグループを作り、今後の作業と役割を明確にした。私は「機械学習組」に配属された。また、「機械学習組」の中でも白線認識プログラム作成の担当となったため、Python と OpenCV 学習の強化と共に白線認識プログラムの先行研究資料の調査なども行った。

### 後期

後期の活動としては主に、白線認識プログラムの作成、白線認識プログラム及び標識認識プログラムと AI カーの制御プログラムの結合、白線認識プログラムの調整、成果発表会の準備を行った。また、「機械学習組」が作成した白線認識プログラムに、「ラジコン組」が作成した AI カーの制御プログラムを組み込んで、「ラジコン組」がその後の調整を行ったため、白線認識プログラムの作成者として中盤からは「ラジコン組」に参加してプロジェクトを進めた。

白線認識プログラムの作成では、先行研究の白線認識などを参考に、AI カー用の白線認識プログラムの作成を行った。しかし、白線認識プログラムは処理が重く、私たちが使用した Raspberry Pi4 では処理が追いつかなかったため、GPU が搭載されている Jetson の使用を検討したが、Jetson の環境構築がうまくいかなかった。そのため、プログラムの軽量化や、処理する画像の画質を落とすなどをして、Raspberry Pi4 での処理を可能にした。また、自分たちの自作コースや AI カーに最適な白線認識プログラムを作るため、白色抽出や逆透視投影変換の調整なども細かく行った。

白線認識プログラムと AI カーの制御プログラムの結合では、「ラジコン組」が作成した PWM 制御などを用いて AI カーのステアリングとスピードを制御するプログラムを受け取り、白線認識プログラムにその制御プログラムを組み込んだ。これにより、AI カーが認識した白線に応じて、AI カーのスピードやステアリングの制御をすることが可能になった。その際には、ステアリングの角度を決定するために必要な制御方法を模索し、PID 制御の使用を検討すると共にそれについての学習を進めた。その後、白線認識の結果から P 制御、PI 制御を利用してステアリングの角度を決定するプログラムを作成した。これによって、AI カーが認識した白線の中心に沿って走ることが可能になった。更に、この白線認識プログラムに標識認識プログラムを組み込むことで AI カーが白線を認識しながら標識を認識することを可能にした。その際には、正しく AI カーが静止、発進するようなプログラムを作成した。具体的には、標識を認識した後に一定時間 AI カーが停止し、その後動き出す際には認識済みの標識を再度認識してしまわないよう無視するプログラムを作成した。

白線認識プログラムの調整では、AI カーが白線の間を滑らかに走行し、自作したコースを自走して周回できるようにするための作業を行った。具体的には、AI カーを実際に走行させながら、PID 制御プログラムの中の比例ゲイン、積分ゲイン、積分時間を微調整して AI カーの自作コース周回を目指した。また、白線認識プログラムを調整するにあたって、パソコンから Raspberry Pi4 を操作する必要があったため、SSH 接続を使ってパソコンから Raspberry Pi4 内のプログラムの実行や書き換えが可能になるような環境を構築し、実際に操作を担当した。

このプロジェクトを通して、仲間と協力して効率よくプロジェクトを進めていく方法を知ることができた。特に、前期のプロジェクトは役割分担とスケジュール管理が疎かであったために、作業の効率が悪く、思ったようにプロジェクトを進めることが出来なかった。しかし、後期は明確な役割分担と細かい目標を設定したスケジュールによって、前期と比較して大きく作業効率が上がり、役割分担とスケジュール管理の大切さを身にしみて理解することができた。そして、この貴重な経験を私の今後の人生に生かしていきたい。

(文責: 芥川裕大)

## 8.6 青柳心吾

### 前期

私はグループ副リーダーとしてリーダーの補佐を行った。日報の発表や意見の出し合いなどを自ら進んで行った。グループの活動としては、テーマの選定、AI カー作成に向けての基礎的な知識、技術の習得に取り組んだ。まず初めに、テーマについてグループで話し合った。私は機械学習に関連したことを行いたく意見を出した。その結果、ラジコンを用いた自動運転となった。次に先行研究を調べた。先行研究には「Donkey Car」があった。また自動運転に必要な画像認識や機械学習関連についても調べグループに共有した。そして、画像認識関連の参考書を購入し、必要となる技術の学習を行いグループに知識を還元した。

個人の活動としては、初めに、基礎的な知識を習得していく中で、画像認識や機械学習にとっても興味が湧いたので、関連の論文や記事で知識を深めた。次に、Python 学習を行った。学習の方法としては、香取教授から頂いた「データサイエンス入門」を利用し Python の基礎を固め、OpenCV や NumPy といった画像認識に必要なライブラリの学習などを行った。

中間発表では役割分担で質疑応答をしたが、知識が不足している事が多く、はっきりと質問に回

## Make Brain Project

答できない事があった。

まとめとして前期はコロナウイルスの影響でオンラインでの活動が多く、調べ学習に多くの時間をかけてしまった。また、情報の取捨選択や共有がうまくいかず、時間を浪費してしまった。そして、グループの人数が7人と多いにも関わらず、役割分担を行っていなかった。そのため、各人の仕事が明確にならず目標を持たずに活動する期間が発生してしまった。これはとても反省しなければいけないと感じた。

### 後期

まず、夏季休暇を利用し前期の大きな反省である、役割分担を行った。ラジコンのスピードやステアリングなどを調節する「ラジコン組」、白線認識や標識認識などを行う「機械学習組」の2つに役割を分けた。そして私は機械学習組となった。次に、グループ内で意見の出し合いが少ないと感じたため、オンラインホワイトボードである「Miro」を使用し、意見を気軽に出せる環境作りを行った。その結果、今まで調べたことや各人の意見を出し合い、活動の方向性が正確になった。また、各自の仕事が明確化し最終発表までの計画を立てる事ができた。

後期ではまず初めに既存の白線認識について学習を行った。白線認識に対しての知識が全くなかったため、Raspberry Pi4のカメラで白線認識を行っているのを参考にした。しかし、精度があまり高くなかったため、使用するROIやIPMや2値化などといった技術を担当に分けて理解を行った。その後、直線やカーブを走らせるためには何が必要であるかを、香取教授に質問しPID制御を教えていただいた。

標識認識については標識の検出と判定が必要であった。まず初めに検出の開発を行なった。検出の方法では、YOLOやSSDなどがあったが処理が重く断念した。そこで処理が軽いカスケード分類器を用いることにした。カスケード分類器を用いた先行研究では、学習に用いる画像の枚数が少なく、ポジティブ画像とネガティブ画像の比率がバラバラであった。また学習に用いる画像がどのような時に精度が向上するのかなど不明な点が多かった。そこで機械学習組の3人で様々な場合のカスケード分類器を作成することで、最も精度がよくなるカスケード分類器の条件を検討した。まずはポジティブ画像とネガティブ画像の比率に注目してカスケード分類器を作成した。しかし、カスケード分類器は学習にとっても時間が必要であったため、3人で情報を共有し効率よく作成することに注意した。その後学習に用いる画像の枚数を増やし、最も精度が高いカスケード分類器を完成させた。

次に標識の判定について、CNNを用いて開発を行なった。しかし、カスケード分類器に多くの時間を使ってしまったため、十分に開発を行う事ができなかった。スケジュールをうまくこなせなかったのが原因である。

成果発表では質疑応答を担当した。聴取者に十分に理解してもらうことを心がけた。

まとめとして、グループの活動では、前期の反省であった役割分担を行い、各自の仕事を明確にすることで開発が大きく進んだ。また開発に対して各人が黙々と作業を行えた。しかし、「ラジコン組」と「機械学習組」と分かれて別々に作業をしてしまったため、情報共有が少なくなってしまう、開発した機能を実際に搭載する際に不明な点が多く出てしまった。これは定期的に進捗状況を共有しなかった事が原因である。また、計画性が見積りが甘く、標識判定を深く開発できなかったのも反省点であった。個人の活動ではPythonやカスケード分類器に関する知識や技術を習得する事ができた。また、自ら積極的に開発や行動を起こす力を身につける事ができた。これらの経験を今後の活動に活かしていきたい。

(文責: 青柳心吾)

## 8.7 折戸啓吾

### 前期

前期の活動では、自動運転に関する基本的な知識の習得や、成果物の制作に必要な物品の選定、Raspberry Pi4 の環境構築などを行った。前期は新型コロナウイルスの影響によってオンラインでの活動が主体であったこともあり、各自で情報収集や調査をすることが主な活動内容であった。しかし、個人個人で学んでいくだけではなく、情報共有をすることでグループ全体として知識を深めていくことができた。また、環境構築においても、グループ内で Raspberry Pi4 を触れたことがあるメンバーはいなかったが、情報を出し合って前期の内に整えることができた。そして、得られた知識や情報を元にどのような成果物を完成させるのかグループ内で話し合いを重ね、大まかな目標を設定することができた。

また、調査を進めていく中で、機械学習で扱うライブラリが豊富にあることや、参考となるプログラムの多くで用いられていることなどから、開発に用いるプログラミング言語を Python に決定した。しかし、前提知識がなかったためプログラミングの学習も並行して行った。担当の香取教授から配布していただいた資料をもとに学習を進めたが、私はプログラミングに苦手意識があったため、配布資料で扱われている内容以外も自主的に学習するように努めた。1、2 年時に講義内で学んだ C 言語や Java と異なる記法に慣れるまで時間を要したが、OpenCV や NumPy といった利用頻度の高いライブラリを中心に、理解を深めていくことができた。

前期の活動における良かった点として、情報収集やプログラミングの学習など自主的に進められることはプロジェクト学習の時間外でも行うことができた。それによって、成果物に関連する記事や先行研究を読み進めることができ、必要最小限の前提知識をスムーズに定着することができたと思う。しかし、反省点として、グループ内で話し合いをする場面において、積極的に発言をしたり意見を提案したりすることができず、他のメンバーに頼ってしまうことが多かった。グループでの活動を進めていく経験を積むためにも、自分の意見をもって活動に参加する意識を持つべきである。話し合いの場ではもちろんのこと、それ以外においてもグループメンバーとコミュニケーションをとるように心がけたい。後期は前期よりオンラインの状況が緩和される可能性があるので、対面で活動できる利点を十分に活かせるように積極的に参加していきたい。全体としては、情報収集に多くの時間を費やしてしまい、前期の内に成果物の制作に着手できなかったことが反省点として挙げられる。情報の集め方やグループでの作業の進め方は前期の活動を通して学ぶことができたので、この経験を後期の活動につなげ、成果物の完成を目指して着実に活動を進めていきたい。

### 後期

後期の活動では、ラジコンの制御を担当するグループと機械学習の実装を担当するグループに分かれて活動を進めていった。私は機械学習のグループに所属した。そして、標識を認識させるためのカスケード分類器の作成を主に担当して行った。前期の活動で環境構築を進める際に、見つけた情報に沿って進めた結果、上手くできなかったことがあった。そのため、後期の活動ではより多くの情報を収集し比較しながら進めていくことを意識した。それにより、自分の意見に自信を持てるようになり、前期と比較すると、積極的に活動に参加することができた。前期の反省点として挙げていたように、人とのコミュニケーションに苦手意識を持っていたので、活動を通して一つ前進できたことであると感じる。カスケード分類器が満足いく精度を得られるまで多くの時間を費やしてしまい、前期に掲げていた目標を大きく変更するという問題も生じたが、作成を繰り返すことで画像認識に関する知識を深められた。

## Make Brain Project

成果発表会の準備をする際、分かれたグループはそれぞれで活動を進めていたため、機械学習に関する原稿を作成するために担当したカスケード分類器に関する内容をまとめた。カスケード分類器の発表内容は専門性が高くなってしまいう可能性があったため、内容をできるだけ細分化して原稿を作成していくことを意識することができた。しかし、成果発表で用いるポスターやスライド、動画などの制作にはグループリーダーや一部のメンバーに作業のほとんどを任せてしまった。我々のグループは他のグループよりも多い人数で活動を行っていたのでその強みを活かせるように、より細かに作業を分担して進めていくべきであったと思う。しかし、発表後の聴講者による評価では良い評価を受けられ、前期よりも良い結果で終わることができたので良かったと思う。

プロジェクト学習を通して、情報収集や知識の習得から始め、自分たちでプログラムを作り、成果物を完成させるという一連の過程を経験することができ、様々な学びを得られた。グループで活動していくからこそ、情報を共有しながら進めていくことが大切であると実感したし、優先順位を決めながら進めていかなければ計画や目標が破綻してしまう恐れもあると改めて感じた。また、作業量に差が生じてしまったり、非効率な進め方をしてしまったり、グループで活動を進めていくことの難しさを体感することもできた。今後グループ単位で活動をする際には今回の反省を振り返り、より充実した活動ができるように貢献していきたい。そして、活動を進めていく中で研究テーマであった自動運転に関する機械学習や画像認識といった技術について知識を深めることもできた。プロジェクト学習の活動を通して得られたこれらの経験を今後の卒業研究などの活動にも生かしていきたい。

(文責: 折戸啓吾)



## 参考文献

- [1] 内閣府. 令和 2 年度 交通事故の状況及び交通安全施策の現況. “道路交通事故の動向”. (Accessed on 5/2021)  
[https://www8.cao.go.jp/koutu/taisaku/r01kou\\_haku/zenbun/genkyo/h1/h1b1s1\\_2.html](https://www8.cao.go.jp/koutu/taisaku/r01kou_haku/zenbun/genkyo/h1/h1b1s1_2.html)
- [2] 川島 賢, 今すぐ試したい! 機械学習・深層学習 (ディープラーニング) 画像認識プログラミングレシビ, 秀和システム, 2019
- [3] Paul Viola and Michael J. Jones: “Rapid Object Detection using a Boosted Cascade of Sample Features”, IEEE CVPR, 2001
- [4] 長谷川修. Local binary pattern, a review. 研究報告グラフィクスとビジュアル情報学 CAD (CG), Vol. 2012, No. 3, pp. 1–6, nov 2012.
- [5] 寺島裕貴, 喜田拓也. 勾配情報を用いた local binary pattern の改良. 第 6 回データ工学と情報マネジメントに関するフォーラム論文集, Vol. 2014, , 3 2014.
- [6] 藤吉弘亘. Hog 特徴量と boosting を用いた人検出. 情報処理学会研究報告. コンピュータビジョンとイメージメディア (CVIM), Vol.2009, 166
- [7] 森のどか, 會澤要, 鈴木拓央, 小林邦和. カスケード型分類器を用いた照明環境変化にロバストなサッカーボール認識. 人工知能学会研究会資料, Vol. 2018, pp. 22- 27
- [8] クジラ飛行机, 杉山陽一, 遠藤俊輔 (2020 年) 「Python による AI・機械学習・深層学習アプリの作り方 TensorFlow2 対応」 ソシム P374~P399
- [9] MOTOYAMA 『PID 制御とは』  
<https://www.motoyama.co.jp/engineer/engi106.htm> (Accessed on 1/2022)
- [10] アズビル株式会社 『PID 制御について』  
[https://www.comclub.com/products/knowledge/jidou\\_seigyoo/jidou\\_seigyoo4.html](https://www.comclub.com/products/knowledge/jidou_seigyoo/jidou_seigyoo4.html) (Accessed on 1/2022)
- [11] TOSHIBA 『PWM とは』  
<https://toshiba.semicon-storage.com/jp/semiconductor/knowledge/e-learning/brushless-motor/chapter3/what-pwm.html> (Accessed on 1/2022)
- [12] ZMP Robot of Everything 『自動運転・ADAS を知る Knowledge』  
[https://www.zmp.co.jp/knowledge/adas\\_dev/adas\\_sensor/adas\\_camera/adas\\_stereo](https://www.zmp.co.jp/knowledge/adas_dev/adas_sensor/adas_camera/adas_stereo) (Accessed on 1/2022)
- [13] ZMP Robot of Everything 『自動運転・ADAS を知る Knowledge』  
<https://www.zmp.co.jp/knowledge/words/01> (Accessed on 1/2022)
- [14] ZMP Robot of Everything 『自動運転・ADAS を知る Knowledge』  
<https://www.zmp.co.jp/knowledge/words/m01> (Accessed on 1/2022)
- [15] HR HAYASHI-REPIC 『超音波センサーとは』  
<https://www.h-repic.co.jp/descriptions/ultrasonic> (Accessed on 1/2022)
- [16] 矢島颯, 高見一正. 自動運転車と手動運転車混在時の進路交譲のための車車間通信プロトコルと試作評価. マルチメディア, 分散, 強調とモバイル (DICOMO2017) シンポジウム, 2017